AD-A094 775 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/4
A HIGHER-ORDER TRAPEZOIDAL VECTOR VORTEX PANEL FOR SUBSONIC FLO--ETC(U)
DEC 80 R E LUTHER
UNCLASSIFIED AFIT/GAE/AA/80D-14 NL

1 OF 1
AD A
094775

END
DATE
FILMED
8-81
DTIC

1

AD A094775

LEVEL II

DDC FILE COPY

UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

81 2 09 088

23 JAN 1981

APPROVED FOR PUBLIC RELEASE AFR 190-17.

Laurel A Lampela

LAUREL A. LAMPELA, 2Lt, USAF
Deputy Director, Public Affairs

Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

DTIC
ELECTE
FEB 1 0 1981
S D
F

A HIGHER-ORDER TRAPEZOIDAL VECTOR
VORTEX PANEL FOR SUBSONIC FLOW

Thesis

AFIT/GAE/AA/80D-14 Ronald E. Luther
Capt USAF

Approved for public release; distribution unlimited.

AFIT/GAE/AA/80D-14

A HIGHER-ORDER TRAPEZOIDAL VECTOR VORTEX PANEL FOR SUBSONIC FLOW

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Ronald E. Luther, B.S.
Capt USAF

Graduate Aeronautical Engineering

December 1980

Preface

I wish to thank my advisor, Major Stephen Koob, for his constant aid and guidance and also tolerance to my efforts. To my family, Freda and Amanda, I can only hope that in the future some reward for their tremendous sacrifice will be granted.

Contents

List of Figures

List of Symbols

α	angle of attack
$\delta(x,y),\gamma(x,y)$	components of vorticity vector
δ^i,γ^i	nodal values of vortex vector
Δ	difference symbol
$\bar{\omega}$	vorticity vector
ϕ_j	set of planform nodal values of vorticity
$[A_{ij}]$	coefficient array
A,B,C,D,E,F	coefficients for δ distribution
G,H,I,J,K,L	coefficients for γ distribution
u,v,w,	perturbation velocities in x, y and z directions
$\hat{i},\hat{j}$	unit vectors in x and y direction
$T^k_{L,T}$	integral of specific region
R	region of integration
U_∞	free stream velocity
M_∞	free stream Mach number
c	chordlength
C_p	pressure coefficient
C_L	local lift coefficient
C_m	local moment coefficient
k	ratio of specific heats
M	number of chordwise panels
N	number of spanwise panels
{ }	vector
[ ]	matrix

Subscripts and Superscripts

L,T   Leading or trailing triangle

i   Node number

Abstract

A higher-order trapezoidal vector vortex panel method is developed for application to linearized subsonic potential flow. Each panel is subdivided into two triangular subregions on which a quadratic vorticity strength distribution is prescribed for both the spanwise and chordwise components of the vorticity vector. The vorticity strength distribution is expressed as a function of the components of the vorticity vector at selected nodes on the boundary of each triangular subregion. Nodal values on the shared boundary of the subregions are made equal, assuring continuity of the vorticity distribution function throughout the trapezoidal panel. A lifting surface of no thickness is modeled with a network of the trapezoidal panels. Again, nodal values on the common panel boundaries are matched to achieve complete continuity of the vorticity distribution throughout the lifting surface. Aerodynamic data for several wing planforms is obtained with the flow model. Results from this method are compared to those from other computational and theoretical methods.

A HIGHER ORDER TRAPEZOIDAL VECTOR VORTEX PANEL FOR SUBSONIC FLOW

I. Introduction

Background

The concept of modeling the flow over a lifting surface by replacing that surface with a distribution of vorticity began in the early part of this century and continues to be expanded and explored. The first quantitative results were achieved by Prandtl in 1919 with his lifting line theory (Ref 9: 112-123) in which the entire surface was represented by a single bound vortex and two infinitely long, free vortices. Despite the many simplifying hypotheses involved in Prandtl's theory, results obtained with it are sufficiently accurate for many purposes and the theory provided the foundation for many subsequent analytical analyses of the lifting surface problem. In 1925 Blenk (Ref 1) extended Prandtl's idea by representing the surface with not one but a distribution of bound vortices over the surface. Blenk's method was an improvement over the single lifting line approach, but still had limitations and the computations involved were very lengthy considering the absence of electronic computers at that time.

A more recent development has been the vortex-lattice theory (Ref 6). This method covers the surface with a grid

of horseshoe vortices and has produced very useful results. Yet another approach, and the one pursued in this report, is to subdivide the surface into a network of panels, each with a discrete vorticity distribution. Such an approach is termed a vortex paneling method.

An accurate flow model via the vortex panel technique is achieved as follows. The vorticity distribution is found such that, at as many points as possible on the surface, the normal component of velocity is zero. This is the so-called kinematic flow condition (Ref 9: 126). With the surface vorticity distribution known, the perturbation velocity at any point on the surface is easily determined. The velocity field and Bernoulli's theorem are then used to compute the pressure distribution and subsequently the aerodynamic lift and moment coefficients.

Ideally, the vorticity vector distribution used to model the lifting surface should be one that resembles the observed physical distribution of vorticity on a finite wing. That is, the vorticity vectors at the wing root lie predominantly in the spanwise direction while those in the region of the wing tip lie predominantly in the chordwise direction. Thus the chosen vorticity distribution function must permit the vorticity to vary in direction. Such a scheme has been proposed by Sparks (Ref 10). His results were flawed, however, until a computer program logic fault was detected by this author. Subsequent results have been encouraging. Sparks' vorticity distribution has two undesirable features. First,

continuity in vorticity distribution is not enforced throughout the surface (Ref 10: 9). This lack of continuity across panel boundaries permits the vorticity distribution at the boundaries to violate Helmholtz's second theorem regarding the continuity of vorticity (Ref 8: 168). Second, the vorticity distribution is linear. It has been shown (Ref 4) that increasing the number of terms in the polynomial expression representing the vorticity distribution has several advantages. The higher order representation reduces surface veloc. y errors and gives significantly improved accuracy as the number of panels used to model the surface is increased.

The present report develops a vortex panel having a quadratic vorticity distribution. The panel is derived specifically to provide a continuous vorticity distribution over the surface while identically satisfying the second Helmholtz condition at every point on the surface.

Approach

The basic problem is developed in Section II. First, the quadratic vorticity distribution function and the importance of the properties of the function are discussed. The panel geometry is presented and unknown values of the components of the vorticity vector are assigned to specified nodes on the panel boundary. This permits the vorticity distribution function to be expressed in terms of the unknown nodal values and the panel geometry. The method of joining panels to form a network to model a wing planform is then

described. This global network has a certain number of nodes and, consequently, a set number of global unknown nodal values to be determined.

Section III describes the solution process. The first step is the application of specific boundary conditions to the leading, tip, trailing and root edges of the planform. This process reduces the total number of unknown nodal values which must be explicitly solved for. The solution is obtained by generating an equal number of linear equations to be solved simultaneously.

Two conditions are satisfied on each panel to generate the required equations. First, the kinematic flow condition is enforced at two control points on each panel. To accomplish this, the normal velocity at each control point must be found. As the normal velocity at any point is affected by the vorticity distribution on the entire planform, each panel's individual contribution to the velocity at any point must be determined. The Biot-Savart Law is applied to the vorticity distribution function of each panel to calculate the induced velocity caused by that panel's vorticity on any desired point in the plane of the panel. The velocities induced by each panel on the desired control point are then summed. This summation yields one equation representing the total induced velocity at one control point. The second condition to be satisfied involves intra-panel continuity of the vorticity distribution. As will be explained in Section II, inter-panel continuity is achieved by the commonality of

nodal values at shared panel boundaries. Such commonality will be shown to not exist, however, on the intra-panel boundary, resulting in a discontinuity along that boundary. This problem is resolved by forcing commonality of sufficient nodal values along the boundary to ensure continuity. Satisfaction of the kinematic flow condition and intra-panel continuity condition results in a system of linear equations. The solution of these gives the nodal values for the vorticity distribution.

Section IV describes how the nodal values are used to calculate the components of the vorticity vector at any point on the planform. From the fully described vorticity distribution, the velocity distribution on the surface of the planform is readily determined and, subsequently, the pressure distribution and aerodynamic coefficients can be computed.

The computer code used to evaluate the theory is outlined in Section V. Section VI presents the results achieved. Comparisons are made to other theories for rectangular wings.

Section VII draws conclusions from the results of this method and offers recommendations for further improvement.

II. Basic Problem Development

Vorticity Distribution Function

The vorticity distribution is a vector function in the x-y plane:

$$\bar{\omega}(x,y) = \delta(x,y)\hat{i} + \gamma(x,y)\hat{j} \tag{2.1}$$

Each component of the vorticity vector is allowed to vary quadratically throughout the plane, thus expressions for the components are:

$$\delta(x,y) = A + Bx + Cy + Dxy + Ey^2 + Fx^2 \tag{2.2}$$

$$\gamma(x,y) = G + Hx + Iy + Jxy + Ky^2 + Lx^2 \tag{2.3}$$

It has been shown (Ref 8: 168) that vorticity is solenoidal and so:

$$\nabla \cdot \bar{\omega} = 0 \tag{2.4}$$

This statement of continuity is known as Helmholtz's second theorem (Ref 8: 168) and also as the condition of source-free vortex distribution (Ref 9: 124).

Equation (2.4) can be satisfied by replacing Eq (2.2) by

$$\delta(x,y) = A - Ix + Cy - 2Kxy + Ey^2 - \tfrac{1}{2}Jx^2 \tag{2.5}$$

Basic Panel Geometry

Figure 1 shows the geometry of the basic trapezoidal panel. This panel is subdivided into a leading triangle and a trailing triangle (hereafter denoted by the subscripts L and T respectively). The basic panel has nine nodes labeled as shown. Two features of the nodal coordinates should be noted. The x-coordinates of nodes 5-9 can be expressed in terms of the x-coordinates of nodes 1-4 (e.g. $x_5 = \frac{1}{2}(x_1+x_2)$). Only two y-coordinates need be specified in advance since $y_3 = \frac{1}{2}(y_1+y_2)$.

Interpolation Functions

It will be shown later that it is advantageous to represent the expressions for δ and γ as functions of nodal values and nodal coordinates. To accomplish this, each triangular subregion of the panel is treated separately. Figure 2 shows the panel split into its subregions with each subregion assigned nine nodal values. Since Eqs (2.5) and (2.3) involve nine coefficients, only nine nodal values are required to uniquely define δ and γ within the subregion. Substituting the nine nodal values of the leading triangle into Eqs (2.3) and (2.5) gives

$$\{\Gamma_L^j\} = [f_i(x_j,y_j)]\{F_L^i\} \qquad (2.6)$$

where

$$\{\Gamma_L^j\} = [\delta_L^1 \ \delta_L^3 \ \delta_L^6 \ \delta_L^7 \ \gamma_L^1 \ \gamma_L^3 \ \gamma_L^4 \ \gamma_L^6 \ \gamma_L^7]^T \qquad (2.7)$$

$$\{F_L^i\} = [A_L \ C_L \ E_L \ G_L \ H_L \ I_L \ J_L \ K_L \ L_L]^T \qquad (2.8)$$

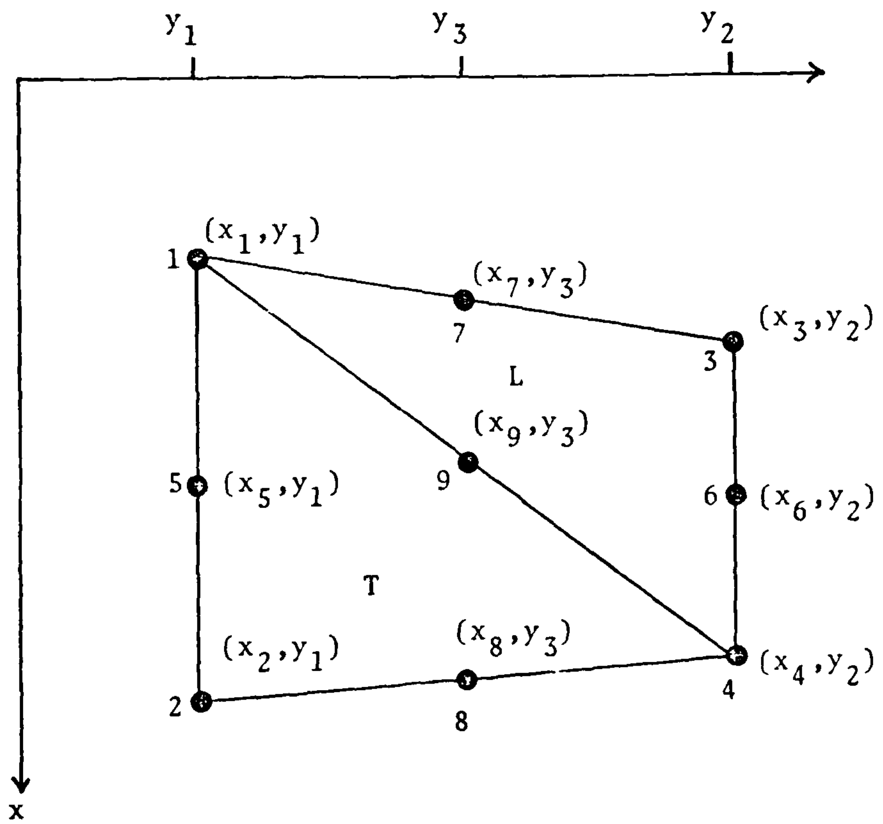

Fig 1. Basic Panel Geometry

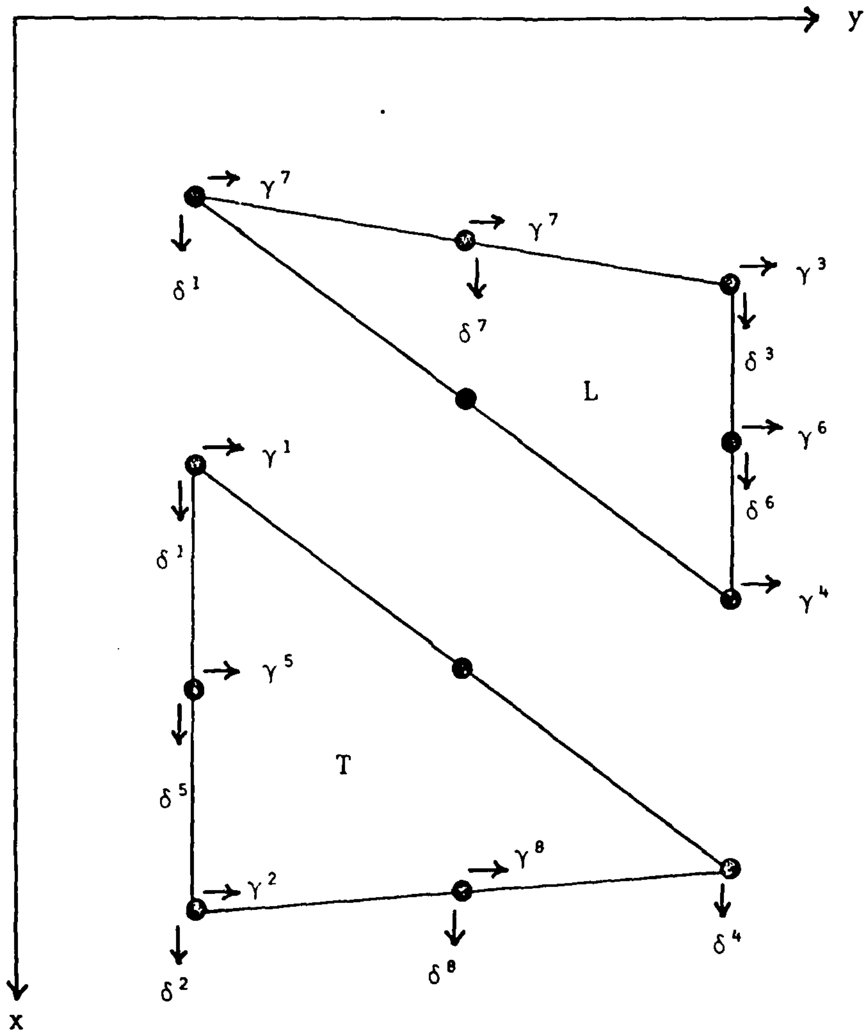

Fig 2. Panel Subregions With Assigned Nodal Values

Similarly for the trailing triangle:

$$\{\Gamma_T^j\} = [g_i(x_j, y_j)]\{F_T^i\} \tag{2.9}$$

where

$$\{\Gamma_T^j\} = [\delta_T^1 \ \delta_T^2 \ \delta_T^4 \ \delta_T^5 \ \delta_T^8 \ \gamma_T^1 \ \gamma_T^2 \ \gamma_T^5 \ \gamma_T^8]^T \tag{2.10}$$

$$F_T^i = [A_T \ C_T \ E_T \ G_T \ H_T \ I_T \ J_T \ K_T \ L_T]^T \tag{2.11}$$

Eqs (2.6) and (2.9) are solved for in the F_L^i and F_T^i, respectively. The method used for obtaining the solutions is explained in Appendix A.

The vorticity vector components can now be written in the following form:

$$\delta_{L,T}(x,y) = [f_{L,T}^i(x,y;\ x_j,y_j)]\{\Gamma_{L,T}^j\} \tag{2.12}$$

$$\gamma_{L,T}(x,y) = [g_{L,T}^i(x,y;\ x_j,y_j)]\{\Gamma_{L,T}^j\} \tag{2.13}$$

In Eqs (2.12) and (2.13) the functions $f_{L,T}^i$ and $g_{L,T}^i$ are the interpolating functions for the components $\delta_{L,T}$ and $\gamma_{L,T}$.

Network Assembly

Figure 3 shows how the panels are connected to model the semi-span of a wing planform. While the planform shown has straight leading and trailing edges, the method also permits analysis of cranked leading and/or trailing edges. The root edge, tip edge and all chordwise boundaries are parallel to the x-axis.

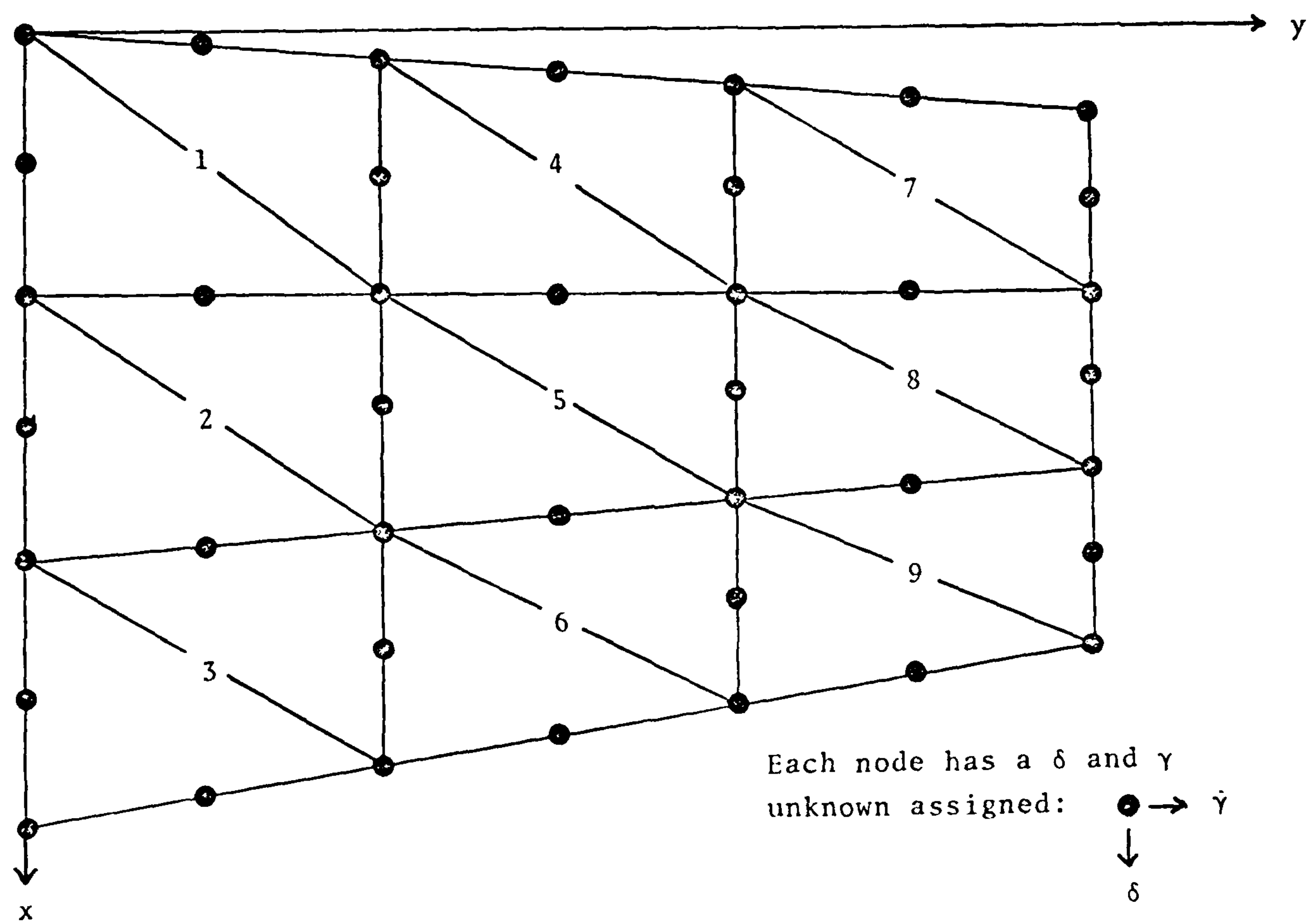

Fig 3. Typical Panel Network with Nodal Unknowns

Inter-Panel Continuity

Continuity of the vorticity function is assured across inter-panel boundaries for the following reason. The line forming a common boundary between two panels is described by

$$y = mx + b \tag{2.14}$$

Let A and B be the two panels whose common boundary is described by Eq (2.14) and examine the value of γ along the boundary for each panel. The value of γ for each panel along the boundary is given by substituting Eq (2.14) into Eq (2.13):

$$\gamma_A\Big|_{\text{Boundary}} = C_1 + C_2x + C_3x^2 \tag{2.15}$$

$$\gamma_B\Big|_{\text{Boundary}} = \overline{C}_1 + \overline{C}_2x + \overline{C}_3x^2 \tag{2.16}$$

The three constants in either Eq (2.15) or Eq (2.16) are determined if three values of γ are specified on the boundary. If the same three values and locations are specified for both Eqs (2.15) and (2.16), then the constants must be identical and γ is continuous along the boundary. The argument can be repeated for the δ function showing it too to be continuous along the boundary.

Intra-Panel Continuity

The same conditions that assure continuity between panels are applied to the shared boundary of the two triangular subregions of the basic panel. Figure 2 shows that continuity is not assured along this boundary since the only nodal values common to both subregions are the δ and γ

components at node 1. Two more common values for both components are required to establish continuity. This can be achieved by enforcing the following conditions. The δ component value at node 4 of the leading triangle is made equal to δ^4. The γ component value at node 4 of the trailing triangle is made equal to γ^4. Finally, the δ and γ component values at node 9 of the leading triangle are made equal to the δ and γ component values at the same node of the trailing triangle. The enforcement of the above conditions assures three common values and locations of both vorticity vector components along the boundary and guarantees intra-panel continuity. The method of incorporating these conditions into the solution process is explained in Section III.

III. Solution Process

Figure 3 shows the problem at hand. A quadratic vorticity vector distribution has been prescribed on the surface of a wing planform of no thickness. The vorticity distribution has been discretized by representing the wing as a network of vortex panels without sacrificing continuity of vorticity anywhere on the wing. By expressing the vorticity distribution in terms of interpolating functions and nodal values, the value of the vorticity vector is determined anywhere on the wing once the nodal values are obtained. This section details the method of solving for the nodal values.

Planform Edge Boundary Conditions

As stated in the introduction, the vorticity distribution has certain observed physical characteristics. These characteristics can be assigned to the model being developed and will serve to simplify the solution process by reducing the number of nodal values that must be determined in order to completely specify the wing's vorticity distribution.

The method used by Cohen (Ref 2) to develop vortex patterns on elliptic wings both with and without sweep is the basis for the boundary conditions that will be imposed here. These boundary conditions are not unique to Cohen's work, with Kuchemann (Ref 5: 140) suggesting similar vortex patterns.

Leading, Tip and Trailing Edges

Figures 4A and 4B illustrate Cohen's straightforward method for deriving the vortex pattern of a tapered wing in straight flight. Figure 4A shows an arbitrary distribution of lift assumed for the wing. Cohen shows the relationship that exists between the pressure distribution and the vortex pattern which yields Fig 4B. The contour lines in Fig 4B correspond to the vorticity pattern of a continuous vortex sheet. Based on these results, the boundary conditions for the leading, tip and trailing edges are formulated.

On the leading edge the vorticity vector is tangent to the leading edge. The slope of the leading edge is given by $\Delta y/\Delta x$, so the vorticity tangency requirement necessitates the δ and γ components at any point on the leading edge be related by the following

$$\frac{\gamma}{\delta} = \frac{\Delta y}{\Delta x}$$

or alternatively

$$\delta = \left(\frac{\Delta x}{\Delta y}\right)\gamma \qquad (3.1)$$

In the solution process, therefore, the δ nodal values on the leading edge can be expressed as functions of the γ components and, consequently, need not be solved for simultaneously.

At the trailing edge, the Kutta condition that no pressure difference exists at the trailing edge requires that the γ nodal values on the trailing edge are identically zero.

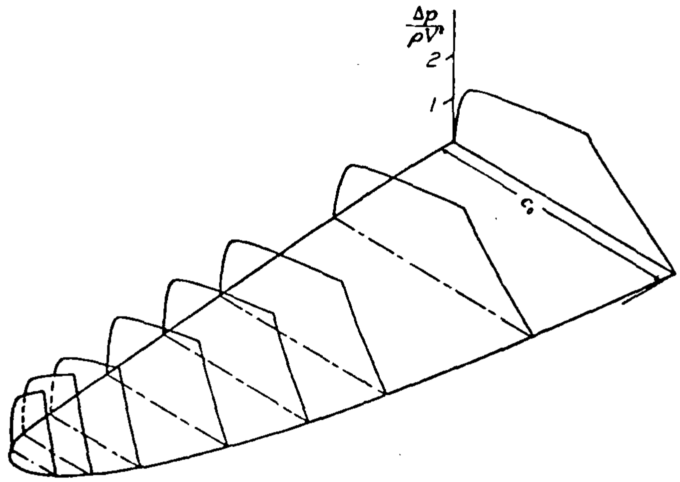

Fig 4A. Lift Distribution (Ref 2)

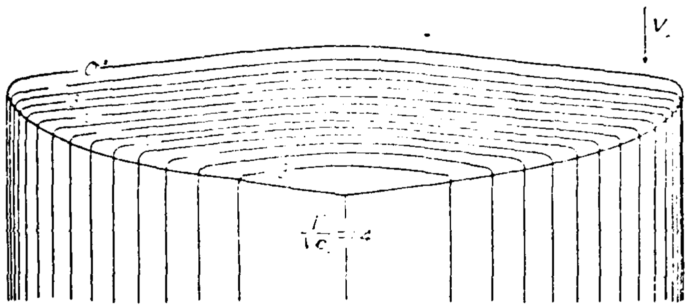

Fig 4B. Vortex Pattern - Straight Wing (Ref 2)

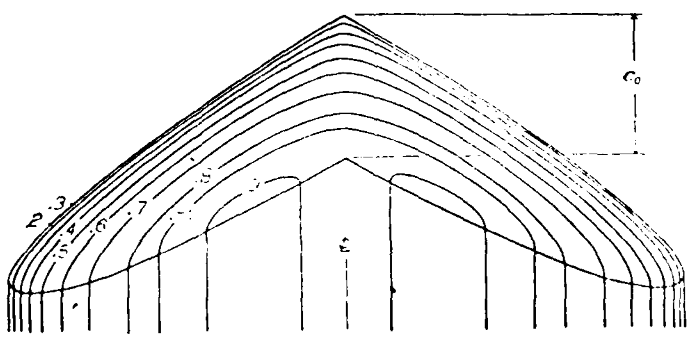

Fig 4C. Vortex Pattern - Swept Wing (Ref 2)

This is because the lift at any point on the surface is proportional to the cross-stream, or γ, component of the vorticity (Ref 2: 544), so requiring no load implies that γ be zero. Similarly, since the pressure differences between the upper and lower wing surfaces decrease to zero toward the wing tips, the γ component of vorticity must also decrease to zero at the tip. This results in the γ nodal values being zero along the tip edge.

Root Edge

Treatment of the root edge is not as straightforward as the leading, tip, and trailing edges. This is especially true for swept wings with pointed apices. Figure 4C shows Cohen's results for the vortex pattern on a sweptback wing. The vortex lines cross the root chord without a discontinuity in slope. It would, therefore, indicate that a boundary condition requiring the δ component of the vorticity vector to be zero along the root chord would be desirable. A conflict develops near the apex, however, where the leading edge boundary condition required that the δ component be non-zero to assure tangency at the leading edge. This dilemma is dealt with as follows. Figure 5A shows the leading edge, root chord, panel or a network modeling a swept wing (node 1 is the apex of the wing). The boundary condition of leading edge tangency is enforced at the apex. At node 2, however, the δ component is made equal to zero, permitting the vorticity vector to cross the root chord without discontinuity in slope. At

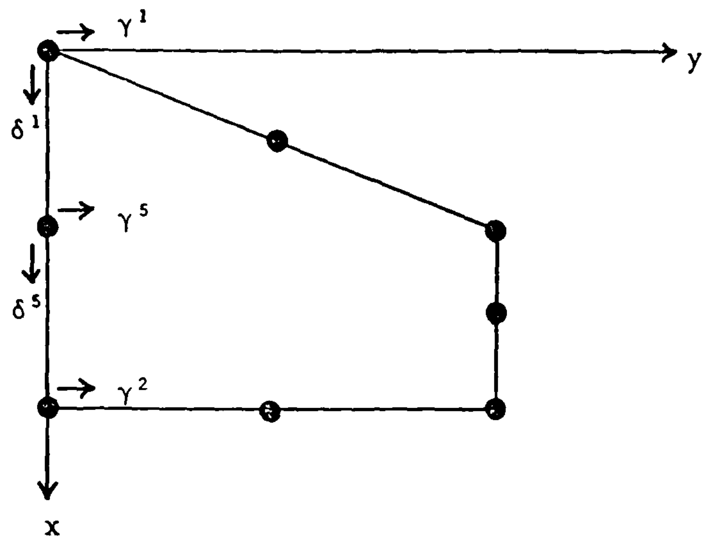

Fig 5A. Leading Edge Root Chord
Panel of Swept Wing

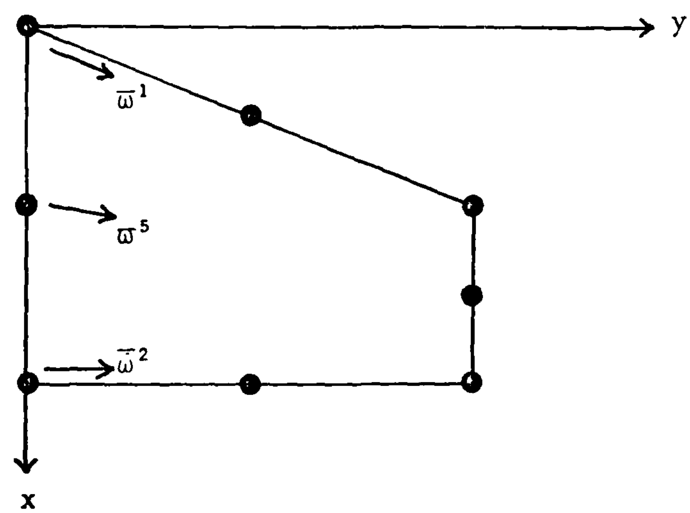

Fig 5A. Boundary Conditions Applied to
Leading Edge Root Chord Panel of Swept Wing

node 5, the δ component is permitted to be finite, but is restricted such that the slope, γ/δ, of the vorticity vector at node 5 is twice that of the vorticity vector at node 1. Figure 5B shows the nature of the vorticity vector, $\bar{\omega}$, on the leading, root chord panel. Because the δ component at node 5 is proportional to the γ component at that node, it does not have to be solved for explicitly. Any other nodes on the root chord due to other panels are treated similarly to node 2.

Applying all edge boundary conditions to the planform of Fig 3 reduces the number of nodal values which must be found to determine the vorticity at any point on the planform. Figure 6 shows the same planform with only the unspecified nodal values numbered. The number of nodal values which must be determined for any network arrangement is 6MN, where M is the number of chordwise panels and N is the number of spanwise panels.

Kinematic Flow Condition

Ideally, the vortex distribution on the lifting surface would result in it being a stream surface such that the normal component of velocity were zero everywhere on that surface. The paneling method as developed here permits the enforcement of only a finite number of boundary conditions so the kinematic flow condition can be satisfied only at certain control points on the surface. In words, the kinematic flow condition states that when the induced velocity at a point on the

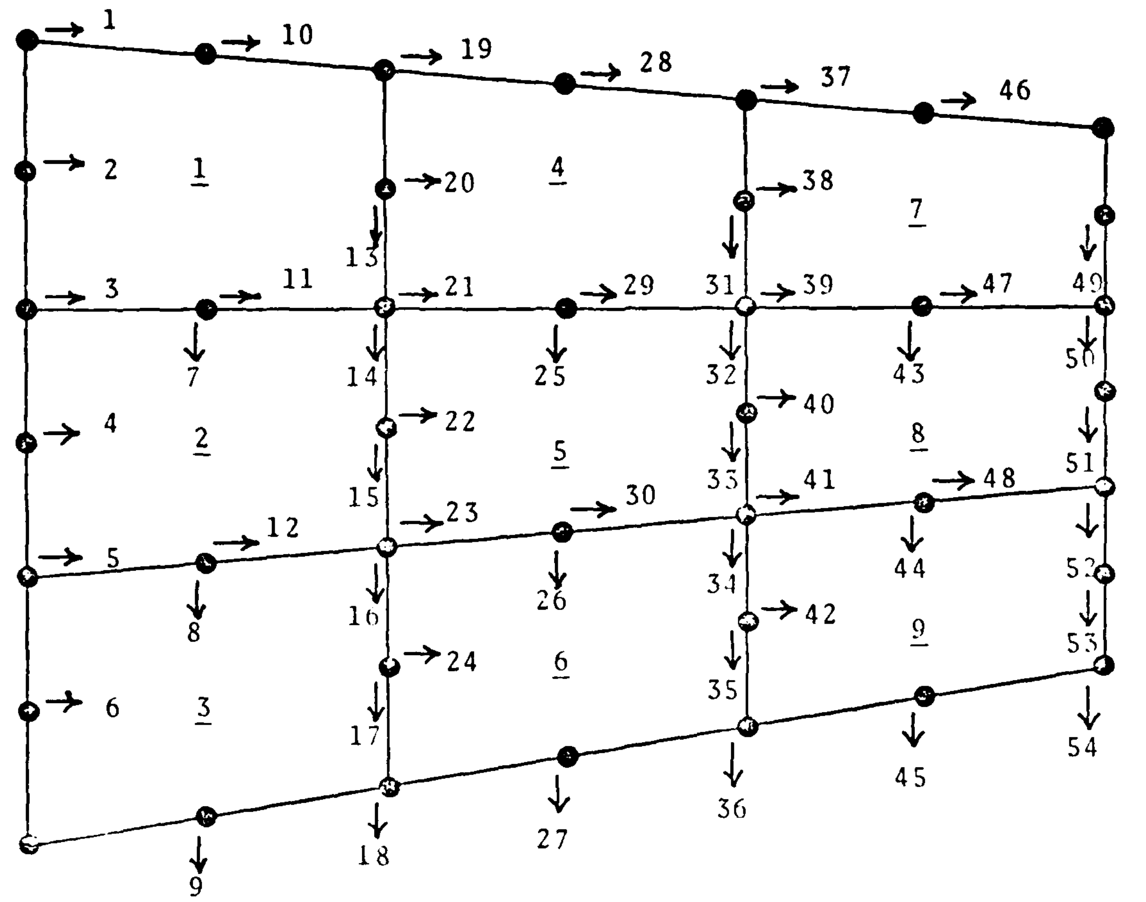

Fig 6. Nine Panel Network With Planform Boundary Conditions Applied

surface, $w(x,y)$, caused by a vortex distribution, $\bar{\omega}(x,y)$, on the surface is added to the normal component of velocity at the point caused by a free stream velocity U_∞ incident to the surface at some angle of attack, α, the resultant velocity is zero. Stated mathematically, the kinematic flow condition is

$$U_\infty \sin \alpha + w(x,y) = 0 \qquad (3.2)$$

The Biot-Savart Law serves to uniquely define the induced velocity coexistent with a given vorticity field (Ref 8: 170). Sparks (Ref 10: 10) has used the Biot-Savart Law and shown that the normal velocity component induced at the origin of an x-y plane, when that plane has a vorticity distribution of the form (2.1), is given by

$$w(o,o) = \frac{1}{4\pi} \int_R \int (x\gamma - y\delta)/(x^2 + y^2)^{3/2} dR \qquad (3.3)$$

Substituting Eqs (2.5) and (2.3) for δ and γ in (3.3) yields:

$$w = \frac{1}{4\pi} \int_R \int (Gx + Hx^2 + 2Ixy - Ay - Cy^2 + 3Kxy^2$$

$$- Ey^3 + Lx^3 + \tfrac{3}{2}Jx^2y)/(x^2 + y^2)^{3/2} dR \qquad (3.4)$$

The induced velocity caused by one quadrilateral panel is the sum of the velocities induced by each of its two triangular subregions. The coefficients in Eq (3.4) are constants over a subregion, they being functions of the nodal values and panel geometry. Eq (3.4) is rewritten as:

$$w = (G_L T_L^1 + H_L T_L^2 + 2I_L T_L^3 - A_L T_L^4 - C_L T_L^5 + 3K_L T_L^6$$

$$- E_L T_L^7 + L_L T_L^8 + \tfrac{3}{2} J_L T_L^9 + G_T T_T^1 + H_T T_T^2 + 2I_T T_T^3 \qquad (3.5)$$

$$- A_T T_T^4 - C_T T_T^5 + 3K_T T_T^6 - E_T T_T^7 + L_T T_T^8 + \tfrac{3}{2} J_T T_T^9)/4\pi$$

where

$$T_{L,T}^k = \int_R \int x^i y^j / (x^2 + y^2)^{3/2} dR \qquad (3.6)$$

and

k =	1	2	3	4	5	6	7	8	9
i =	1	2	1	0	0	1	0	3	2
j =	0	0	1	1	2	2	3	0	1

The evaluation of the first five integrals given by Eq (3.6) is found in Sparks (Ref 10: 47-53). The last four were evaluated in a similar manner; the results are given in Appendix B. With the integrals evaluated in terms of the panel geometry, Eq (3.5) is expressed in the form:

$$w = \sum_{i=1}^{8} (f_i \delta^i + g_i \gamma^i) \qquad (3.7)$$

where the f_i and g_i are expressions involving only the panel geometry.

Eq (3.7) gives the normal velocity induced at the origin of the x-y plane by a vorticity vector distribution over a trapezoidal panel in the plane.

Intra-Panel Continuity Condition

Intra-panel continuity is achieved by enforcing the

four conditions set forth in Section II. The first of these is

$$\delta^4 = \delta_L(x_4, y_2) \tag{3.8}$$

Substituting Eq (2.12) for the right-hand side of Eq (3.8) gives an expression in terms of only nodal values and panel geometry. The second condition involves matching the γ values at node 4 and yields:

$$\gamma^4 = \gamma_T(x_4, y_2) \tag{3.9}$$

The right-hand side of Eq (3.9) is replaced by Eq (2.13), giving an expression in terms of nodal values and panel geometry.

The final two conditions match the δ and γ values at node 9:

$$\delta_T(x_9, y_3) = \delta_L(x_9, y_3) \tag{3.10}$$

$$\gamma_T(x_9, y_3) = \gamma_L(x_9, y_3) \tag{3.11}$$

Eqs (2.12) and (2.13) are substituted into the left- and right-hand sides, respectively, of Eqs (3.10) and (3.11).

Eqs (3.8) through (3.11) involve only nodal values and panel geometry, and satisfying these four expressions assures intra-panel continuity of the vorticity distribution function.

Eq (3.9) cannot be applied to the trailing edge root chord panel (panel 3 in Fig 6) under all conditions. If the trailing edge of this panel is parallel to the y-axis, Eq (3.9) is trivial. The reason for this can be seen by examining Figs 2 and 6. Suppose the panel in Fig 2 is the

trailing root chord panel. Then, by the planform edge boundary conditions, the following are all identically zero: δ^1, δ^5, δ^2, γ^2, γ^8, and γ^4. The value of γ on the trailing edge is

$$\gamma_T = C_1 + C_2 y + C_2 y^2 \tag{3.12}$$

since x is constant on the edge. Two values of γ, (γ^2 and γ^8), are specified. Helmholtz's second theorem, Eq (2.4), says that:

$$\frac{\partial f}{\partial x} + \frac{\partial \gamma}{\partial y} = 0 \tag{3.13}$$

Since δ^1, δ^2 and δ^5 are all zero, $\frac{\partial \delta}{\partial x}$ at node 2 is zero, therefore $\frac{\partial \gamma}{\partial y}$ must also be zero at that node. Eq (3.12) is completely specified if the three conditions

$$\gamma^2 = \gamma^8 = \frac{\partial \gamma}{\partial y} = 0 \tag{3.14}$$

are dictated. The solution is that γ is zero along the line. Consequently, trying to apply Eq (3.9), recalling that γ^4 is also zero, is redundant. This problem is avoided by always insuring that the trailing edge of this panel (root-trailing edge) has a non-zero slope so that the edge is never parallel to the y-axis.

Matrix Formulation

Eq (3.7) is the induced velocity at a point caused by one panel. The total induced velocity at a point is found by summing the effects of all panels in the planform. The effect of panels on the planform to the left of the x-axis

is accounted for by reflecting the control point about the x-axis and applying Eq (3.7) to that point. Figure 7 illustrates this procedure. The induced velocity at point A caused by panels 1 and 2 is desired. The effect of panel 1 is found by straightforward application of Eq (3.7). From symmetry, the induced velocity at point A caused by panel 2 is identical to the effect of panel 1 on point A'. So, the induced velocity at point A due to panels 1 and 2 is the sum of the effect of panel 1 on A and the effect of panel 1 on A'. For a selected control point, the induced velocity is given by:

$$\sum_{j=1}^{6MN} A_{ij}\phi_j = w_i \tag{3.15}$$

In Eq (3.15) the coefficients A_{ij} are functions of panel geometry only, and the ϕ_j are the unknown nodal values of the vorticity vector distribution. The right-hand side of Eq (3.12) is the normal component of velocity such that Eq (3.2) is satisfied:

$$w_i = -U_\infty \sin \alpha_i \tag{3.16}$$

Two control points are selected per panel so Eq (3.15) gives 2MN equations for the 6MN unknown nodal values.

The remaining 4MN equations come from the four intra-panel continuity conditions being applied to each panel. The resulting system of equations has the form:

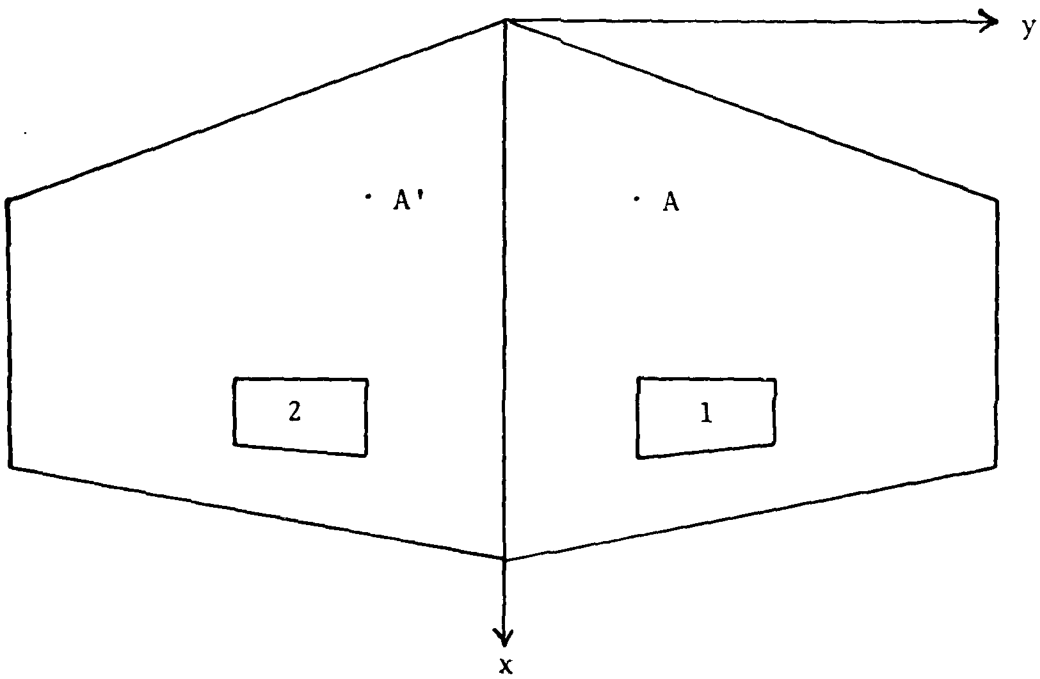

Fig 7. Control Point Reflection

$$[A_{ij}] \begin{Bmatrix} \phi_1 \\ \vdots \\ \phi_{6MN} \end{Bmatrix} = -U_\infty \begin{Bmatrix} \sin \alpha_1 \\ \vdots \\ \sin \alpha_{2MN} \\ \hline 0 \end{Bmatrix} \tag{3.17}$$

The solution to this system is:

$$-\begin{Bmatrix} \dfrac{\phi_j}{U_\infty} \end{Bmatrix} = [A_{ij}]^{-1} \begin{Bmatrix} \sin \alpha_i \\ \hline 0 \end{Bmatrix} \tag{3.18}$$

<u>Compressibility Correction</u>

The Prandtl-Glauert Rule is used to account for the effect of compressibility as the Mach number is increased (Ref 8: 276). In practice, the adjustment for compressibility is made by multiplying all x-coordinates in the A matrix of Eq (3.18) by the factor

$$\beta = 1/\sqrt{1 - M_\infty^2} \tag{3.19}$$

IV. Aerodynamic Data

The ϕ_j, as determined from Eq (3.15), are used in Eqs (2.12) and (2.13) to compute the components of the vorticity vector at any point on the planform. The perturbation velocities on the surface due to the vorticity distribution are:

$$\bar{u} = \pm \frac{1}{2}\gamma\hat{i} \tag{4.1}$$

$$\bar{v} = \mp \frac{1}{2}\delta\hat{j} \tag{4.2}$$

where the upper and lower signs correspond to the upper and lower sides of the surface (Ref 9: 124).

Following Sparks, the perturbation velocities are used to determine the pressure coefficient by one of two methods. The exact isentropic expression is (Ref 11: 433):

$$C_p = 2[1+([k-1]M_\infty^2/2)(1-[(U_\infty+u)^2 + v^2]/U_\infty^2)^{k/(k-1)} - 1]/KM_\infty^2 \tag{4.3}$$

while the second order approximation is (Ref 11: 433):

$$C_p = -2[2u/U_\infty+(1-M_\infty^2)u^2/U_\infty^2+v^2/U_\infty^2] \tag{4.4}$$

Let the difference in the pressure coefficients of the lower and upper surfaces be defined by:

$$\Delta C_p = C_{p_L} - C_{p_u} \tag{4.5}$$

The local coefficients of lift and moment are then given by (Ref 9: 30):

$$C_L = \frac{1}{C} \int_0^C \Delta C_p \, dx \qquad (4.6)$$

and

$$C_M = -\frac{1}{C^2} \int_0^C (\Delta C_p) x \, dx \qquad (4.7)$$

V. Computer Code

The theory set forth in Sections II, III and IV has been incorporated into FORTRAN code. The program, WING2, takes its basic structure from the program WING, developed in Ref 10. WING2's mainline and two of its subroutines, MESH and LOADS2, are the result of only minor modifications to their counterparts in WING. Each subroutine is briefly described below and a complete listing of WING2 is in Appendix C.

WING2: Reads data describing the planform to be analyzed. Calls all subroutines.

MESH: Computes the x-y coordinates of the nodes and control points for the input planform.

BIOT: Generates the 2MN equations which result from enforcing the kinematic flow condition at each control point.

INT: Evaluates the nine integrals given by Eq (3.6).

STRIP: Is called by INT and contains expressions for evaluating the integrals.

CONT: Generated 4MN equations which result from assuring intra-panel continuity.

COEFF: Evaluates the coefficients required to define the interpolating functions.

LOADS2: Uses the nodal values of the vorticity distribution to determine the pressure distribution and aerodynamic coefficients.

VI. Results and Discussion

The theory was evaluated by obtaining aerodynamic data for a rectangular wing of aspect ratio 5 at an angle of attack of 5°. Two panel networks were used. First, the wing semispan was modeled with one panel. Second, four panels of equal size were used.

Figure 8A shows the distribution of the local lift coefficient, C_L, over the span using the one panel model and varying the control point location. Figure 8B shows the control point set locations on the semispan. The results shown for these locations illustrate the sensitivity of the solution to the control point placement. Analysis of results for the four control point locations shown and others indicated that control point set A was the most desirable.

Figure 9 shows the C_L distribution for control point set A compared to the distribution predicted by Truckenbrodt's lifting surface theory (Ref 9: 164). Although WING2 consistently underpredicted C_L values, the distribution is smooth and exhibits the desired rate of decay from a maximum at the root to zero at the tip.

Figure 10A illustrates the value of the vorticity vector at various points on the semispan, again for the single panel model. Also plotted on the figure are the two control point locations. The nature of the vortex pattern agrees

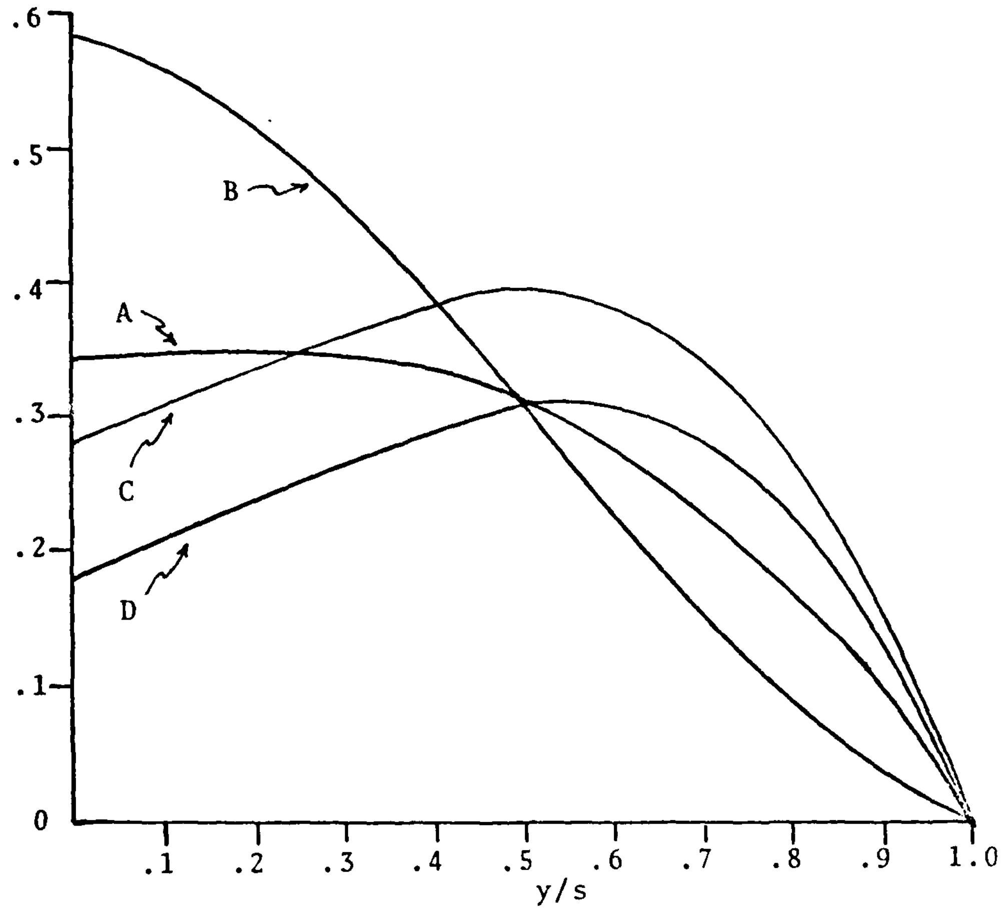

Fig 8A. Lift Distribution Versus Span Station for Various Control Point Locations

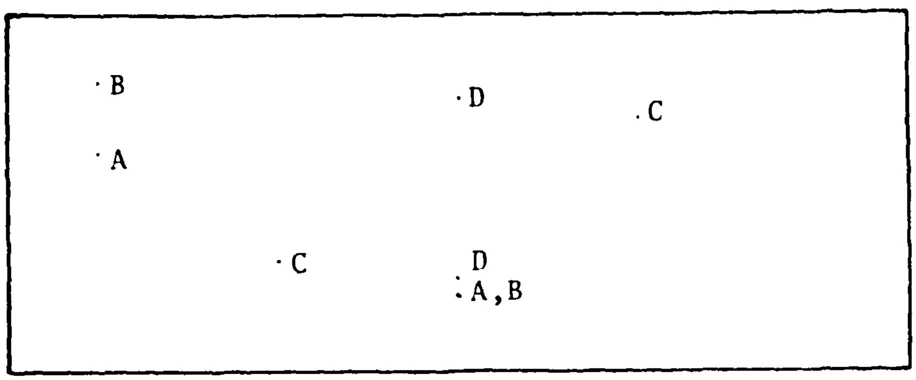

Fig 8B. Control Point Locations

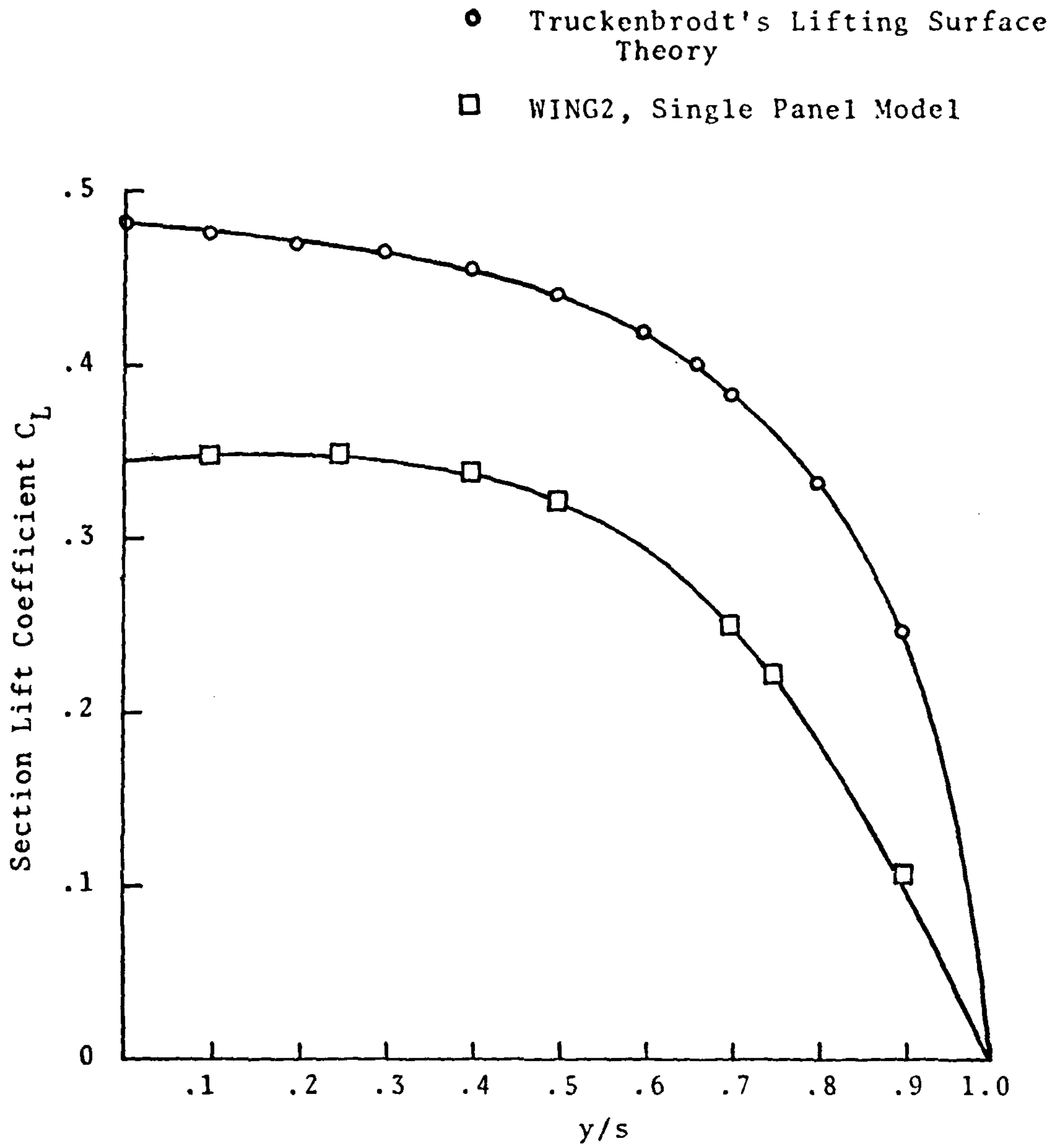

Fig 9. Lift Distribution Versus Span Station for Rectangular Wing; AR=5, α=5°

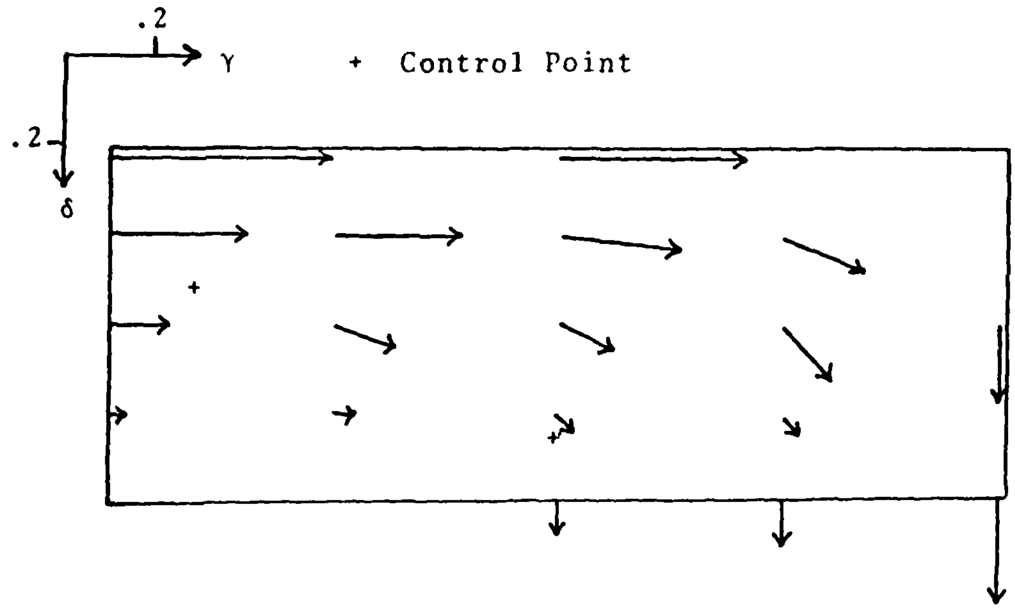

Fig 10A. Vortex Pattern - Single Panel Model

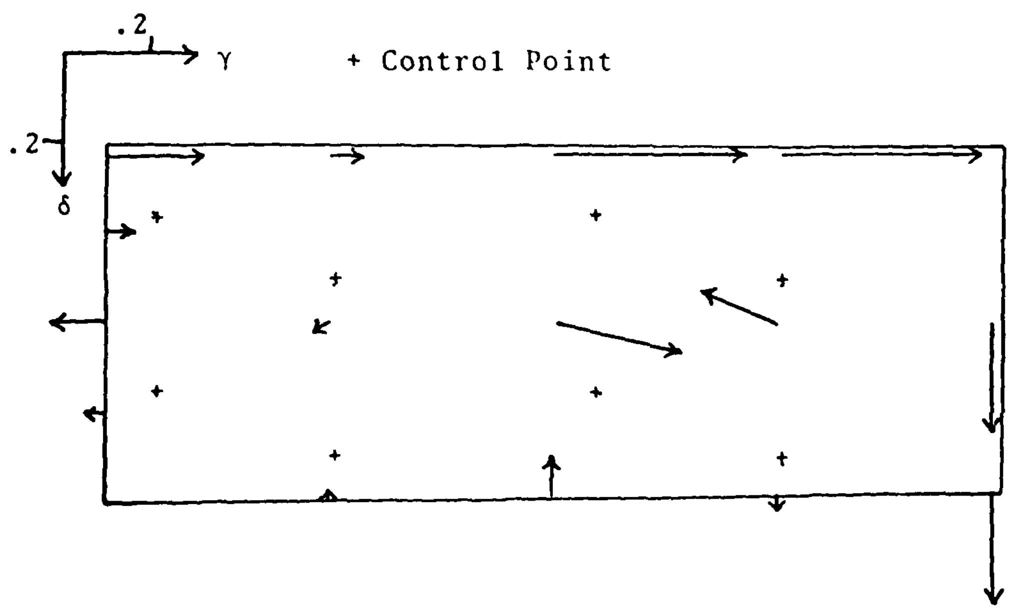

Fig 10B. Vortex Panel - Four Panel Model

with the desired pattern, Fig 4B.

Figure 11 plots the center of pressure versus span station. The X_{CP} shifts slightly forward towards the tip instead of slightly rearward as desired (Ref 9: 159).

The desirable features exhibited by the one panel model all but vanish when four panels are used. Figure 10B graphically shows the altered nature of the vortex pattern. The prevalence of negative values of both δ and γ is not physically reasonable for this low α case. A further illustration of the departure of the four panel solution is shown in Fig 12. Here the chordwise distribution of γ is given along the root chord. Three distributions are shown; the one panel model, four panel model, and the exact answer for two-dimensional flow about a flat plate (Ref 5: 62):

$$\gamma(x) = 2\alpha\left(\frac{x_L - x}{x - x_T}\right) \qquad (6.1)$$

where x_L and x_T are the values of the leading and trailing edges respectively. For the single panel model, WING2 is more correct in the midchord region than in either the leading or trailing edge regions.

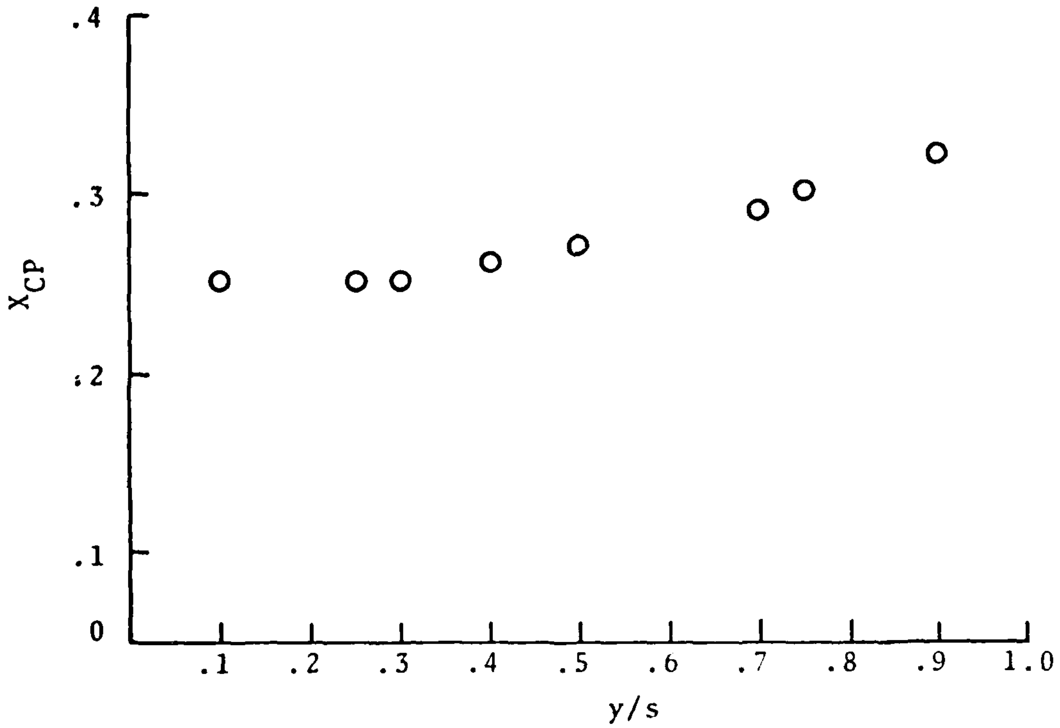

Fig 11. X_{CP} Versus Span Station for Rectangular Wing; AR=5, α=5°

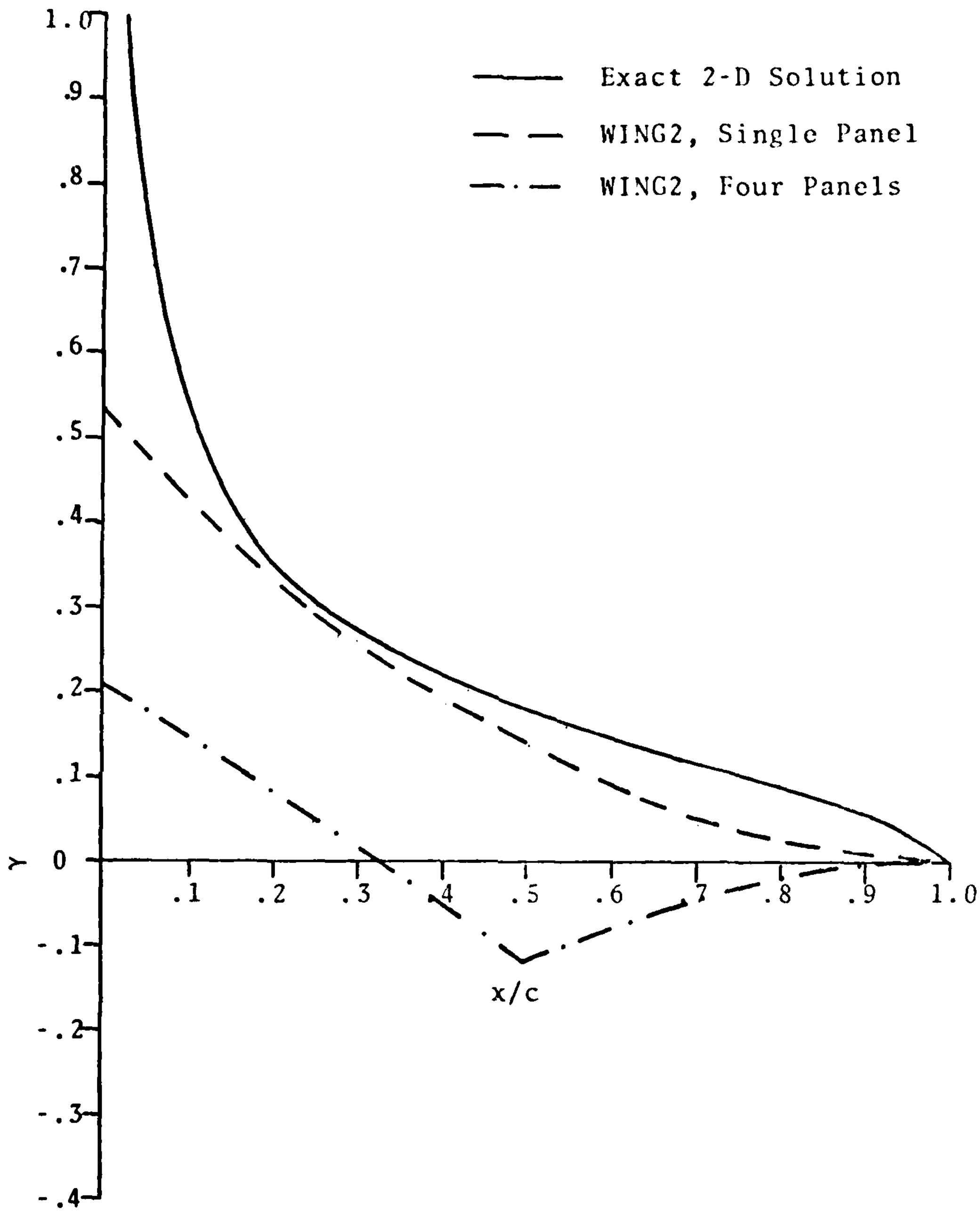

Fig 12. Comparison of γ Distribution
Along Root Chort

VII. Conclusions and Recommendations

The linear discontinuous vorticity vector method of Sparks (Ref 10) has been extended to a quadratic, continuous vector vortex panel method. The desirable features of the higher-order, continuous vorticity distribution are apparent from the results of the single panel model. The spanwise and chordwise lift distributions, while consistenly low in value, do exhibit the characteristics of the desired solutions. The method breaks down, however, if the planform is modeled with four panels.

No reason is readily apparent for the drastic change in the nature of the vortex pattern for the four panel solution. It is suspected that the vorticity distribution function must be further restricted so that the solution converges toward a physically reasonable result.

The following are recommendations which may improve the results of the method.

a. Sacrifice some control point equations to provide continuity in the first derivative of γ with respect to y across panel boundaries.

b. Incorporate analytical results into the solution process. Davies (Ref 3) has demonstrated an accurate closed form approximation to the behavior of the lift distribution near the wing apex. Using the analytical result will insure

a more correct result in the apex area and may then improve the overall lift distribution.

c. Adopt an iterative procedure. Starting with the single panel solution, use selected values of δ and γ as given quantities in the four panel solution process.

Bibliography

1. Blenk, Hermann. "The Monoplane as a Lifting Vortex Surface," National Advisory Committee for Aeronautics, Technical Memorandum No. 1111. Washington, D.C., 1947.

2. Cohen, Doris. "A Method for Determining the Camber and Twist of a Surface to Support a Given Distribution of Lift, With Applications to the Load Over a Sweptback Wing," National Advisory Committee for Aeronautics, Technical Report No. 826. Washington, D.C., 1945.

3. Davies, Patricia J. "The Load Near the Apex of a Lifting Swept Wing in Linearized Subsonic Flow," Royal Aircraft Establishment Technical Report 72031, January 1972.

4. DeJongh, J.E. and Kooh, S.J. "An Evaluation of Higher Order Aerodynamic Paneling Methods for Two-Dimensional Incompressible Flow," AIAA Paper 78-14, January 1978.

5. Kuchemann, D. The Aerodynamic Design of Aircraft. London: Pergamon Press, 1978.

6. Langley Research Council. Vortex-Lattice Utilization. National Aeronautics and Space Administration, SP-405, 1975.

7. The Mathlab Group. MACSYMA Reference Manual. Laboratory for Computer Science, Massachusetts Institute of Technology, 1977.

8. Milne-Thompson, L.M. Theoretical Aerodynamics. New York: Dover Publications Inc., 1973.

9. Schlichting, Hermann and Truckenbrodt, Erich. Aerodynamics of the Airplane. New York: McGraw-Hill Book Co., 1979.

10. Sparks, John C. "Development and Application of a Subsonic Triangular Vortex Panel." Unpublished MS thesis (AFIT/GAE/AE/80J-1). Air Force Institute of Technology, Wright-Patterson AFB, Ohio, June 1980.

11. Yuan, S.W. Foundations of Fluid Mechanics. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1967.

APPENDIX A

Interpolating Function Equations

Interpolating Function Equations

Eqs (2.5) and (2.3) are transformed into functions of nodal values and nodal coordinates. The solutions to Eqs (2.7) and (2.10) are

$$\{F_L^i\} = [f_i(x_j, y_j)]^{-1} \{\Gamma_L^j\} \tag{A.1}$$

and

$$\{F_T^i\} = [g_i(x_j, y_j)]^{-1} \{\Gamma_T^j\} \tag{A.2}$$

The f_i and g_i matrices were symbolically inverted using the computer program MACSYMA (Ref 7). With the symbolic inversion the $F_{L,T}^i$ are readily expressed as linear polynomials in the nodal values. For example,

$$\begin{aligned} F_L^1 = A_L = {} & (ALD1)(\delta_L^1) + (ALD3)(\delta_L^3) + (ALD6)(\delta_L^6) \\ & + (ALD7)(\delta_L^7) + (ALG1)(\gamma_L^1) + (ALG3)(\gamma_L^3) \\ & + (ALG4)(\gamma_L^4) + (ALG6)(\gamma_L^6) + (ALG7)(\gamma_L^7) \end{aligned} \tag{A.3}$$

where the ALDX and ALGX represent the appropriate element of the inverted f_i matrix. The complete algebraic expressions of the inverted f_i and g_i matrices are found in the subroutine COEFF listed in Appendix C.

APPENDIX B

Evaluation of Integrals

Evaluation of Integrals

The procedure for evaluating the four integrals in Eqs (3.11) - (3.14) is given in detail in Sparks (Ref 10: 47-53). Only the results are given here.

Following Sparks' notation, the STRIP functions for the four additional integrals are:

$$S_6[(x_0,y_0),(x_1,y_1)] = \{(\frac{my_1}{2(m^2+1)} + \frac{3bm}{2(m^2+1)^2})$$

$$\cdot\ (m^2y_1^2 + m^2x_1^2)^{1/2} - (\frac{my_0}{2(m^2+1)} + \frac{3bm}{2(m^2+1)})$$

$$\cdot\ (m^2y_0^2 + m^2x_0^2)^{1/2} + \frac{3bm^2 - m(m^2+1)b^2}{2(m^2+1)^{5/2}}$$

$$LN(\frac{(m^2+1)^{1/2}(y_1^2+x_1^2)^{1/2}+my_1+x_1}{(m^2+1)^{1/2}(y_0^2+x_0^2)^{1/2}+my_0+x_0})\} \qquad (B.1)$$

$$S_7[(x_0,y_0),(x_1,y_1)] = \{(\frac{2b-y_1}{2(m^2+1)} - \frac{3b}{2(m^2+1)^2})$$

$$\cdot\ (m^2y_1^2 + m^2x_1^2)^{1/2} + (\frac{2b+y_0}{2(m^2+1)} + \frac{3b}{2(m^2+1)^2})$$

$$\cdot\ (m^2y_0^2 + m^2x_0^2) - \frac{3m^2b^2}{2(m^2+1)^{5/2}}$$

$$LN(\frac{(m^2+1)^{1/2}(y_1^2+x_1^2)^{1/2}+my_1+x_1}{(m^2+1)^{1/2}(y_0^2+x_0^2)^{1/2}+my_0+x_0})\} \qquad (B.2)$$

$$S_8[(x_0,y_0),(x_1,y_1)] = \{(\frac{mx_1(m^2+1)-m^2b+2b}{2(m^2+1)^2} - y_1)$$

$$\cdot\ (x_1^2+y_1^2)^{1/2} + (y_o - \frac{mx_0(m^2+1)-m^2b+2b}{2(m^2+1)^2})$$

$$\cdot\ (x_0^2+y_0^2)^{1/2} + \frac{3mb^2}{2(m^2+1)^{5/2}}$$

$$LN(\frac{(m^2+1)^{1/2}(x_0^2+y_0^2)^{1/2}+my_0+x_0}{(m^2+1)^{1/2}(x_1^2+y_1^2)^{1/2}+my_1+x_1})\} \tag{B.3}$$

$$S_9[(x_0,y_0),(x_1,y_1)] = \{(\frac{3mb+m^4x_1+m^2x_1}{2(m^2+1)^2})$$

$$\cdot\ (y_1^2+x_1^2)^{1/2} - (\frac{3mb+m^4x_0+m^2x_0}{2(m^2+1)^2})$$

$$\cdot\ (y_0^2+x_0^2)^{1/2} + \frac{y_0^2}{2}LN(x_0+(x_0^2+y_0^2)^{1/2})$$

$$- \frac{y_1^2}{2}LN(x_1+(x_1^2+y_1^2)^{1/2}) + \frac{2m^2b^2-b^2}{2(m^2+1)^{5/2}}$$

$$LN(\frac{(m^2+1)^{1/2}(y^2+x^2)^{1/2}+my_0+x_0}{(m^2+1)^{1/2}(y_1^2+x_1^2)^{1/2}+my_1+x_1})\} \tag{B.4}$$

The expressions for the four integrals are then

$$T_L^i = S_i[(x_1,y_1),(x_3,y_2)] - S_i[(x_1,y_1),(x_4,y_2)] \tag{B.5}$$

$$T_T^i = S_i[(x_1,y_1),(x_4,y_2)] - S_i[(x_2,y_1),(x_4,y_2)] \tag{B.6}$$

APPENDIX C

Computer Code

```
      PROGRAM WINC2 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)
      COMMON/BLOCK1/X(  ,4),Y(  ,2),XD(  ),YD(  ),Z(  ,2),CY,CX,DYA,DYB,
     1DYA,DXB,XCA(  ),YCA(  ),XCB(  ),YCB(  )
      COMMON/BLOCKB/NS,SSPN,S(  ),  ,C(  ),  ,NL(  ),B(  ,  ),MACH
      COMMON/BLOCKC/Z(  ,  ),SS(  ),  (  ),SUM(  ),  (  ),  
      COMMON/BLOCKD/IL(3),IT(3)
      COMMON/BLOCKE/  (  ,3),KILL
      COMMON/BLOCKF/  ,CLTOT(  ),  
      COMMON/BLOCKK/G1,G2,G3,G4,G5,G6,G7,G8,D1,D2,D3,D4,D5,D6,D7,D8
      COMMON/BLOCKL/ALD1,ALD2,ALD3,ALD7,ALG1,ALG3,ALG5,ALG7,ALG8,
     1             DLD1,DLD2,DLD3,DLD7,DLG1,DLG3,DLG5,DLG7,DLG8,
     2             ELD1,ELD2,ELD3,ELD7,ELG1,ELG3,ELG5,ELG7,ELG8,
     3             ILD3,ILD4,ILG1,ILG3,ILG4,ILG5,ILG7,
     4             JLD3,JLD4,JLG1,JLG3,JLG4,JLG5,JLG7,
     5             KLD3,KLD4,KLG1,KLG3,KLG4,KLG5,KLG7,
     6             GLD3,GLD4,GLG1,GLG3,GLG4,GLG5,GLG7,
     7             HLD3,HLD4,HLG1,HLG3,HLG4,HLG5,HLG7,
     8             LLG3,LLG4,LLG5,
     9             ATD1,ATD2,ATD3,ATD5,ATD  ,ATG1,ATG2,ATG5,ATG8,
     A             CTD1,CTD2,CTD3,CTD5,CTD  ,CTG1,CTG2,CTG5,CTG8,
     B             ETD1,ETD2,ETD3,ETD5,ETD  ,ETG1,ETG2,ETG5,ETG8,
     C             ITD1,ITD2,ITD3,ITG1,ITG2,ITG5,ITG8,
     D             JTD1,JTD2,JTD3,
     E             KTD1,KTD2,KTD3,KTG1,KTG2,KTG5,KTG8,
     F             GTD1,GTD2,GTD3,GTG1,GTG2,GTG5,GTG8,
     G             HTD1,HTD2,HTD3,HTG1,HTG2,HTG5,HTG8,
     H             LTG1,LTG2,LTG5
      REAL  ILD3,ILD4,ILG1,ILG3,ILG4,ILG5,ILG7,
     1      JLD3,JLD4,JLG1,JLG3,JLG4,JLG5,JLG7,
     2      KLD3,KLD4,KLG1,KLG3,KLG4,KLG5,KLG7,
     3      LLG3,LLG4,LLG5,
     4      ITD1,ITD2,ITD3,ITG1,ITG2,ITG5,ITG8,
     5      JTD1,JTD2,JTD3,
     6      KTD1,KTD2,KTD3,KTG1,KTG2,KTG5,KTG8,
     7      LTG1,LTG2,LTG5
      DIMENSION WKAREA(24 ),PB(1)
      REAL MACH
      REAL IL,IT
      INTEGER G1,G2,G3,G4,G5,G6,G7,G8,D1,D2,D3,D  ,D5,D6,D7,D8
C
C                BEGIN INPUT SEQUENCE FOR GEOMETRIC DATA.
C                ALL DATA IS READ USING FREE FORMAT.
C                READ SPAN DATA.
C
      READ(5,*) KILL
      READ(5,*) SSPN,NS,(S(I),I=1,NS)
      WRITE(6,1  )
 1    FORMAT(////  X,*SPAN STATIONS*/25X,*YLOC*,11X,*Y/S*)
      DO 1  I=1,NS
 1    WRITE(6,11 ) I,S(I)
 11   FORMAT(  X,I  ,  X,F  .3)
      WRITE(6,12 ) SSPN
 12   FORMAT(/2 X,*SEMI-SPAN =*,F7.2)
C                READ CHORD DATA.
      READ(5,*) NC,(C(I),I=1,NC)
      WRITE( ,13 )
 13   FORMAT(//2 X,*CHORD STATIONS*/25X,*XLOC*,11X,*X/C*)
```

```
      DO 2  I=1,NC
    2 WRITE(6,14) I,C(I)
   14 FORMAT(21X,I3,10X,F .3)
C                         READ BREAK POINT DATA.
      READ(5,*) NB,((NL(I),B(I,1),B(I,2)),I=1,NB)
      WRITE(6,15)
   15 FORMAT(///20X,*BREAK POINTS*/22X,*XLE*,1 X,*TE*,10X,*YLOC*)
      DO 3  I=1,NB
    3 WRITE(6,16) B(I,1),B(I,2),NL(I)
   16 FORMAT(1 X,F .3,3X,F .3,3X,I3)
C                         READ CONTROL POINT LOCATIONS AND MACH
      READ(5,*) CY,CY,CYA,CXA,CYB,CXB,MACH
      WRITE(6,17) MACH,CY,CY,CYA,CXA,CYB,CXB
   17 FORMAT(//20X,*MACH NUMBER =*,F5.3///31X,*CY =*,F5.3/31X,
     1*CY =*,F5.3/31X,*CYA =*,F5.3/31X,*CXA =*,F5.3/31X,*CYB =*,F5.3/31X
     1,*CXB =*,F .3)
C                         READ THE NUMBER OF LOADING CASES TO BE
C                         EXAMINED AND THE CALCOMP PLOTTING OPTION.
C                         ENTER 0 TO SKIP THE PLOT ROUTINES, OR ENTER
C                         1 IF PLOTS ARE DESIRED.  THE ANGLE OF ATTACK
C                         AND CAMBER DISTRIBUTION FOR EACH CASE WILL
C                         BE READ LATER IN THE PROGRAM.  THIS IS DONE
C                         TO PRESERVE STORAGE SPACE.
      READ(5,*) NA,KPLOT
      WRITE (6,18) NA
   18 FORMAT (//,20X,*NO. OF LOADING CASES =*,I3)
      IF (KPLOT .EQ. 0) GO TO 20
      WRITE (6,19)
   19 FORMAT (///25X,*PLOTS REQUESTED*)
      GO TO 22
   20 WRITE (6,21)
   21 FORMAT (///25X,*NO PLOTS DESIRED*)
   22 CONTINUE
C                         CALCULATE M AND N. M IS THE NUMBER OF
C                         CHORDWISE PANELS AND N IS THE NUMBER
C                         OF SPANWISE PANELS
      M=NC-1
      N=NS-1
C                         CALCULATE NUMBER OF PANELS NP.
      NP=M*N
C                         CALL SUBROUTINES
      CALL MESH(NP)
      IF(KTLL.EQ.0) GO TO 27
      IF(KTLL.EQ.1) GO TO 999
   27 NP2=2*NP
      CALL DICI (M,N,NP,NP2)
      CALL GDNI (M,N,NP)
      IJOB=1
      IA=120
      DL=-1.
      CALL LINV3F (A,B2,IJOB,NP2,IA,D1,D2,WKAREA,IER)
      DO 500 I=1,NP
      WRITE (6,100)(A(I,J),J=1,9)
      WRITE (6,990)(A(I,J),J=10,18)
      WRITE (6,990)(A(I,J),J=19,27)
      WRITE (6,990)(A(I,J),J=28,36)
      WRITE (6,990)(A(I,J),J=37,40)
```

```
      WRITE (5,599)(A(I,J),J=1,5)
  599 FORMAT (5(2X,E12.5))
      WRITE (5,597)
  597 FORMAT(///)
  598 CONTINUE
      CALL LOADSZ(NP,M,I,IP2)
  999 STOP
      END
```

```
      SUBROUTINE INT(X1,X2,X3,X4,Y1,Y2)
C                   THIS SUBROUTINE EVALUATES THE PANEL INTE-
C                   GRALS. IT ALSO FUNCTIONS AS THE EXECUTIVE
C                   CONTROL ROUTINE FOR SUBROUTINE STRIP.
      COMMON/BLOCKB/IL(3),IT(3)
      COMMON/BLOCKL/SPF(9,3),KILL
      REAL IL,IT
C                   CALCULATE STRIP FUNCTIONS FOR THE PANEL
C                   LEADING EDGE WITH CORNER POINTS (X1,Y1)
C                   AND (X3,Y2).
      I=1
      CALL STRIP(X1,Y1,X3,Y2,I)
C                   CALCULATE STRIP FUNCTIONS FOR THE PANEL
C                   MAIN DIAGONAL WITH CORNER POINTS (X1,Y1)
C                   AND(X4,Y2).
      I=2
      CALL STRIP(X1,Y1,X4,Y2,I)
C                   CALCULATE STRIP FUNCTIONS FOR THE PANEL
C                   TRAILING EDGE WITH CORNER POINTS (X2,Y1)
C                   AND (X4,Y2).
      I=3
      CALL STRIP(X2,Y1,X4,Y2,I)
C                   EVALUATE PANEL INTEGRALS.
      DO 1  I=1,3
      IL(I)=SPF(I,1)-SPF(I,2)
      IT(I)=SPF(I,2)-SPF(I,3)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STRIP(X.,Y.,X1,Y1,I)
C                    SUBROUTINE TO CALCULATE STRIP FUNCTIONS.
C                    THERE ARE 15 STRIP FUNCTION CALCULATIONS
C                    PER QUADRILATERAL PANEL.
      COMMON/BLOCKE/SPF(9,3),KILL
      REAL M.,MO,BO
C                    USE STRIP FUNCTION EQUATIONS FOR AN IN-
C                    FINITE SLOPE IF X1=X..
      IF(ABS(X1-X.).LT.1E-6) GO TO 1..
      M.=(Y1-Y.)/(X1-X. )
      BO=Y  -X *M
      DUM1=SQRT(X. **2+Y. **2)
      DUM2=SQRT(X1**2+Y1**2)
      DUM3=SQRT(1+M.**2)
      DUM4=(DUM1+(X  +M  *Y  )/DUM3)/(DUM2+(X1+M.*Y1)/DUM3)
      DUM4=ABS(DUM4)
      DUM4=ALOG(DUM4)
      DUM5=ABS(DUM1+X.)
      DUM5=ALOG(DUM5)
      DUM6=ABS(DUM2+X1)
      DUM6=ALOG(DUM6)
      SPF(1,I)=-[illegible]*DUM4/DUM3
      SPF(2,I)=Y [illegible]DUM5-Y1[illegible]DUM6-[illegible](DUM1-DUM2)/DUM3**2-B[illegible]DUM4/DUM3**3
      SPF(3,I)=M [illegible](DUM3-DUM1)/DUM3**2-B[illegible]*M [illegible]DUM4/[illegible]**3
      SPF(4,I)=DUM4/DUM3+DUM5-DUM6
      SPF(5,I)=-[illegible]**2*DUM4/DUM3**3+MO*(DUM1-DUM2)/DUM3**2
      MO=M.
      BO=B
      SPF(6,I)=((MO*Y1)/2./DUM3**2+(3.*BO*MO)/2./DUM3**4)*MO*DUM2-((MO*
     1Y.)/2./DUM3**2+(3.*BO*MO)/2./DUM3**4)*MO*DUM1-DUM4*(2.[illegible]**3)**2-BO
     1**2-B[illegible]/2./DUM3**5
      SPF(7,I)=[illegible]DUM2*((-Y1+2.*BO)/2./DUM3**2-3.*BO/2./DUM3**4)+[illegible]DUM1
     1*((Y -2.*BO)/2./DUM3**2+3.*BO/2./DUM3**4)-DUM4*(3.*MO**2*BO**3/2./
     1DUM3**5)
      SPF(8,I)=(3.*[illegible]**2)*DUM4/(2.*DUM3**5)+DUM2*(([illegible]Y1*DUM3**2
     1-MO*BO+2.*BO)/(2.*DUM3**4)-Y1)+DUM1*(Y -([illegible]DUM3-[illegible]**2*BO
     1+2.*BO)/(2.*DUM3**4))
      SPF(9,I)=(([illegible]**2*[illegible]**2-BO*[illegible]2)/(2*DUM3**5)) DUM4+(([illegible]-MO**2*
     1Y1+M[illegible]2*X1)/([illegible]DUM3**4))*DUM2+((-[illegible] Y -M[illegible]X [illegible]*BO)/
     2(2*DUM3**4))*DUM1+(Y **2/2)*DUM5-(Y1**2/2)*DUM6
      WRITE *,"SPF(9,I)=",SPF(9,I)
   [illegible] RETURN
 1[illegible] CONTINUE
      DUM1=SQRT(X **2+Y **2)
      DUM2=SQRT(X **2+Y1**2)
      DUM3=ABS((DUM1+Y  )/(DUM2+Y1))
      SPF(1,I)=-ALOG(DUM3)
      DUM3=ABS(DUM1+Y.)
      DUM3=ALOG(DUM3)
      DUM4=ABS(DUM2+Y1)
      DUM4=ALOG(DUM4)
      SPF(2,I)=Y [illegible]DUM3-Y1*DUM4
      SPF(3,I)=[illegible]2-[illegible]M1
      SPF(4,I)=[illegible]-[illegible]M[illegible]
      SPF(5,I)=-Y [illegible]SPF(1,I)
      DUM5=ABS(Y.+[illegible]1)
      DUM6=ABS(Y1+DUM2)
```

```
DUM7=DUM5/DUM6
 DUM7 =ALOG(DUM7)
SPF(6,I)=(Y1*DUM3)/2.-(Y2*DUM1)/2.+((X2**2)/2.)*DUM7
WRITE*,"SPF(6,I)=",SPF(6,I)
SPF(7,I)=X2*DUM1-Y2*DUM2
WRITE*,"SPF(7,I)=",SPF(7,I)
SPF(8,I)=-Y1*DUM2+Y2*DUM1
WRITE*,"SPF(8,I)=",SPF(8,I)
SPF(9,I)=(X1*DUM3)/2-((Y1**2)/2)*DUM4-(X2*DUM1)/2+((Y2**2)/2)*DUM7
WRITE*,"SPF(9,I)=",SPF(9,I)
RETURN
END
```

```
      SUBROUTINE MESH (NP)
C               SUBROUTINE TO CALCULATE PANEL CORNER POINTS
C                     TS AND CONTROL POINTS.
      COMMON/BLOCKA/X([illegible],9),Y([illegible],3),XC([illegible]),YC([illegible]),E([illegible],2),CY,CX,CYB,CXB,
     1CYA,CXA,XC1([illegible]),YC1([illegible]),XC3([illegible]),YC3([illegible])
      COMMON/BLOCKB/NS,SSPN,S(1[illegible]),NC,C(1[illegible]),NR,NL(1[illegible]),B(1[illegible],2),MACH
      COMMON/BLOCKE/AREA,CLTOT(1[illegible]),NPLOT
      NO=1
      NR1=NR-1
      NC1=NC-1
      DO 4  K=1,NR1
      DO 1  I=1,NC
      E(I,1)=B(K,1)+C(I)*(B(K,2)-B(K,1))
    1 E(I,2)=B(K+1,1)+C(I)*(B(K+1,2)-B(K+1,1))-E(I,1)
      ID1=NL(K)
      ID2=NL(K+1)
      ID3=ID2-1
      DO 3  J=ID1,ID3
      D2=S(ID2)-S(ID1)
      D3=S(ID1)
      DO 2  I=1,NC1
      X(NO,1)=E(I,1)+E(I,2)*(S(J)-D3)/D2
      X(NO,2)=E(I+1,1)+E(I+1,2)*(S(J)-D3)/D2
      X(NO,3)=E(I,1)+E(I,2)*(S(J+1)-D3)/D2
      X(NO,4)=E(I+1,1)+E(I+1,2)*(S(J+1)-D3)/D2
      X(NO,5)=(X(NO,1)+X(NO,2))/2.
      X(NO,6)=(X(NO,3)+X(NO,4))/2.
      X(NO,7)=(X(NO,1)+X(NO,3))/2.
      X(NO,8)=(X(NO,2)+X(NO,4))/2.
      X(NO,9)=(X(NO,5)+X(NO,6))/2.
      Y(NO,1)=S(J)*SSPN
      Y(NO,2)=S(J+1)*SSPN
      Y(NO,3)=(Y(NO,1)+Y(NO,2))/2.
      NO=NO+1
    2 CONTINUE
    3 CONTINUE
    4 CONTINUE
C                     CALCULATE CONTROL POINTS.
      DO 5  I=1,NP
      YC(I)=(1-CY)*Y(I,1)+CY*Y(I,2)
      XC(I)=CY*X(I,3)+(1-CY)*X(I,1)+CX*(CY*(X(I,4)-X(I,3))+(1-CY)*
     1(X(I,2)-X(I,1)))
      YC1(I)=(1-CYA)*Y(I,1)+CYA*Y(I,2)
      YC3(I)=(1-CYB)*Y(I,1)+CYB*Y(I,2)
      XC1(I)=CYA*X(I,3)+(1-CYA)*X(I,1)+CXA*(CYA*(X(I,4)-X(I,3))+(1-CYA)*
     1(X(I,2)-X(I,1)))
    5 XC3(I)=CYB*X(I,3)+(1-CYB)*X(I,1)+CXB*(CYB*(X(I,4)-X(I,3))+(1-CYB)*
     1(X(I,2)-X(I,1)))
C                     PRINT A TABLE CONSISTING OF THE PANEL NO.
C                     AND THE CORRESPONDING CORNER AND CONTROL
C                     POINTS.
      WRITE(6,1[illegible])
  1[illegible] FORMAT (////1X,*PANEL*,[illegible]X,*X1*,[illegible]X,*X2*,[illegible]X,*X3*,[illegible]X,*X4*,[illegible]X,*X[illegible]*,
     1[illegible]X,*X[illegible]*,[illegible]X,*X7*,[illegible]X,*X8*,[illegible]X,*X9*,[illegible]X,*Y1*,[illegible]X,*Y2*,[illegible]X,*Y3*)
      DO 6  I=1,NP
      WRITE (6,[illegible])  I,X(I,1),X(I,2),X(I,3),X(I,4),X(I,5),X(I,6),X(I,7),
     1X(I,[illegible]),X(I,9),Y(I,1),Y(I,3),Y(I,2)
```

```
1??   FORMAT (1X,I2,12(3X,F7.2))
 6    CONTINUE
      WRITE (6,102)
1??   FORMAT (////1X,'PANEL',7X,'XO',8X,'YO',9X,'XOA',6X,'YOA',3X,
     1'XOB',8X,'YOB')
      DO 51 I=1,NP
      WRITE(6,103) I,XO(I),YO(I),XOA(I),YOA(I),XOB(I),YOB(I)
1??   FORMAT (2X,I2,6(3X,F7.2))
 5?   CONTINUE
C
C                    CALCULATE WING PLANFORM AREA
      AREA=0.
      DO 70 L=1,N1
         B1=R(L,2)-R(L,1)
         B2=R(L+1,2)-R(L+1,1)
         N1=JL(L)
         N2=JL(L+1)
         H=(S(N2)-S(N1))*.5*SPAN
         AREA=AREA+H*(B1+B2)
 7    CONTINUE
      WRITE (6,7?) AREA
 7    FORMAT(///2?X,'TOTAL WING AREA =',F8.2)
C
      RETURN
      END
```

```
      SUBROUTINE NODNUM (M,N,I)
      COMMON/BLOCKK/G1,G2,G3,G4,G5,G6,G7,G8,D1,D2,D3,D4,D5,D6,D7,D8
      INTEGER G1,G2,G3,G4,G5,G6,G7,G8,D1,D2,D3,D4,D5,D6,D7,D8
      INTEGER CRN,SRN
C
C           THIS SUBROUTINE COMPUTES GLOBAL UNKNOWN
C           NODAL VALUES APPLICABLE
C           NODAL VALUES APPLICABLE TO A GIVEN PANEL I.
C
C           COMPUTE THE CHORDWISE ROW NUMBER (CRN)
C           AND SPANWISE ROW NUMBER (SRN) FOR PANEL
C           I, (INTEGER ARITHMETIC USED)
C
      CRN=(I-1)/M+1
      SRN=I-(CRN-1)*M
C
      L=(CRN-1)*3*M
C
      G1=L+2*SRN-1
C
      G2=G1+2
C
      G3=G1+3*M
C
      G4=G3+2
C
      G5=G1+1
C
      G6=G7+1
C
      D8=L+2*M+SRN
C
      D7=D8-1
C
      G8=D8+M+1
C
      G7=G8-1
C
      D5=G1-2*M
C
      D1=D5-1
C
      D2=D5+1
C
      D3=G1+4*M-1
C
      D6=D3+1
C
      D4=D3+2
C
      RETURN
      END
```

```
      SUBROUTINE COEFF (X1,X2,X3,Y.,Y1,Y2)
      COMMON/BLOCKL//ALD1,ALD3,ALD5,ALD7,ALG1,ALG3,ALG5,ALG6,ALG7,
     1               CLD1,CLD3,CLD5,CLD7,CLG1,CLG3,CLG5,CLG6,CLG7,
     2               ELD1,ELD3,ELD5,ELD7,ELG1,ELG3,ELG5,ELG6,ELG7,
     3               ILD3,ILD5,ILG1,ILG3,ILG5,ILG6,ILG7,
     4               JLD3,JLD5,JLG1,JLG3,JLG5,JLG6,JLG7,
     5               KLD3,KLD5,KLG1,KLG3,KLG5,KLG6,KLG7,
     6               GLD3,GLD5,GLG1,GLG3,GLG5,GLG6,GLG7,
     7               HLD3,HLD5,HLG1,HLG3,HLG5,HLG6,HLG7,
     8               LLG3,LLG5,LLG6,
     9               ATD1,ATD2,ATD3,ATD4,ATD5,ATG1,ATG2,ATG3,ATG5,
     A               CTD1,CTD2,CTD3,CTD4,CTD5,CTG1,CTG2,CTG3,CTG5,
     B               ETD1,ETD2,ETD3,ETD4,ETD5,ETG1,ETG2,ETG3,ETG5,
     C               ITD1,ITD2,ITD3,ITG1,ITG2,ITG3,ITG5,
     D               JTD1,JTD2,JTD3,
     E               KTD1,KTD2,KTD3,KTG1,KTG2,KTG3,KTG5,
     F               GTD1,GTD2,GTD3,GTG1,GTG2,GTG3,GTG5,
     G               HTD1,HTD2,HTD3,HTG1,HTG2,HTG3,
     H               LTG1,LTG2,LTG5
      REAL  ILD3,ILD5,ILG1,ILG3,ILG5,ILG6,ILG7,
     1      JLD3,JLD5,JLG1,JLG3,JLG5,JLG6,JLG7,
     2      KLD3,KLD5,KLG1,KLG3,KLG5,KLG6,KLG7,
     3      LLG3,LLG5,LLG6,
     4      ITD1,ITD2,ITD3,ITG1,ITG2,ITG3,ITG5,
     5      JTD1,JTD2,JTD3,
     6      KTD1,KTD2,KTD3,KTG1,KTG2,KTG3,KTG5,
     7      LTG1,LTG2,LTG5
      ATD1      = ((X2**2+X1*Y2)*Y2**2+((-7*Y2-X1)*Y1+X2**2+3*X1*X2)*Y1*Y
     1  2+(5*Y1**2+(-X2-3*X1)*X )*Y1**2)/(((Y2**2-2*(1*Y2+X1**2)*Y2**2+(
     2  -2*Y2**2+4*Y1*Y2-2*Y1**2)*Y1*Y2+(X2**2-2*Y1*X2+X1**2)*Y1**2)
      ATD2      = ((X1*Y2+(1**2)*Y2**2+((-3*X2-3*Y1)*X1+[illegible]*Y2**2+[illegible](1*Y2+Y
     1  1**2)*Y1*Y2+([illegible]*X1**2+(5*Y2-3*X1)*Y1)*Y1**2)/(((X2**2-2*(1*Y2+Y1-
     2  *2)*Y2**2+(-2*Y2**2+4*X1*Y2-2*X1**2)*Y1*Y2+(X2**2-2*(1*Y2+X1**2
     3  )*Y1**2)
      ATD4      = (Y1*Y2+Y1**2)/(Y2**2-2*Y1*Y2+Y1**2)
      ATD5      = -(([illegible]*X1*Y2*Y2**2+((-1[illegible]*X2-4*Y1)*X1-([illegible]*X2**2+12*Y1-X2)*Y1*
     1  Y2+(12*X1**2+([illegible]*X2-12*X1)*X1)*Y1**2)/(((Y2**2-2*Y1*Y2+Y1**2)*Y2*
     2  *2+(-2*Y2**2+[illegible]*X1*X2-2*Y1**2)*Y1*Y2+(Y2**2-2*Y1*X2+X1**2)*Y1**2
     3  )
      ATD8      = -4*Y1*Y2/(Y2**2-2*Y1*Y2+Y1**2)
      ATG1      = (([illegible]X2*Y1**2+(-[illegible]X2**2-4*X1*X2) Y1+[illegible]*Y1*Y2**2)*Y1*Y2+(-
     1  4*Y1**3+([illegible]*Y2+[illegible])*X1**2-[illegible]X1*X2*X1)*Y1**2)/(((X2**2-2*X1*Y2+(1
     2  **2)*Y2**3+(-[illegible]*Y2**2+[illegible]*Y1*Y2-3*Y1**2)*Y1*Y2**2+(7*Y2**2-5*Y1*X2
     3  +3*Y1**2)*Y1*[illegible]2*Y2+(-X2**2+2*Y1*X2-X1**2)*Y1**3)
      ATG2      = ((([illegible]*Y2*Y1**2+([illegible]*X2**2-12*X1+X2)[illegible]-[illegible]*Y1*Y2**2+3*X1**2*Y
     1  2)*Y1*Y2+(-[illegible]*Y[illegible]+(12*X1-[illegible]X2)*Y[illegible]+([illegible]Y1*X2-[illegible]Y1**2)*X1) Y1
     2  *2)/((X2**2-2 Y1*Y2+X1**2)*Y2**3+(-[illegible]Y1**2+[illegible]Y1*X2-3*Y1**2) Y1
     3  Y2**2+(3*X2**2-[illegible]*Y1*X2+3*Y1**2)*Y1**2*Y2+(-X2**2+2*X1*X2-Y1**2)
     4  *Y1**3)
      ATG3      = -(([illegible]*Y2+Y[illegible]**2-1[illegible]*Y1*Y2*X[illegible]-[illegible]*Y2**2+1[illegible]*Y1*Y2**2)*Y1*Y2+(-
     1  [illegible]Y1**3+1[illegible]Y1[illegible]2+([illegible]*Y2[illegible]-15*(1*Y2)*Y1)*Y1**2) /((X2**2-2*Y1*
     2  X2+X1**2)*Y2**3+(-3*X2**2+[illegible]Y1*X2-3*Y1**2)*Y1*Y2**2+(3*X2**2-3*
     3  X1*X2+3*Y1**2) Y1**2*Y2+(-Y2**2+2*Y1*X2-X1**2)*Y1**3)
      ATG8      = -([illegible]*X2*Y1*Y2-[illegible]*Y[illegible] Y1**2)/(Y2**3-3*Y1*Y2**2+3*Y1**2*Y2-Y
     1  1**3)
      CTD1      = (((7*Y2+Y1)*Y[illegible]-7*X2**2-5*Y1*X2)*Y2+(-12*X[illegible]**2+([illegible]*Y2*7*Y
     1  1)*X[illegible]-X2**2-3*X1*Y2)*Y1)/(((X2**2-2*Y1*X2+Y1**2)*Y2**2+(-2 Y2**
```

```
2    +.*X1*X2-2*Y1**2)*Y1*Y2+(X2**2-2*X1 Y2+Y1**2)*Y1**2)
 DTD2     = (((  *X2+3*X1)*X2+.*Y2**2-3 X1*X2-7*Y1**2)*Y2+(-12*Y [illegible]
1    +(21*X1-  *Y2)*Y  +4*Y2**2-7*X1*X2-X1**2) Y1)/((Y2**2-2*Y1*Y2+Y1*
2    *2)*Y2**2+(-2*X2**2+ *Y *X2-2*X1**2)*Y1*Y2+(X2**2-2*X1*X2+X1**2
3    )*Y1**2)
 DTD4     = -(Y2+3*Y1)/(Y2**2-2*Y1*Y2+Y1**2)
 DTD5     = -(((12*X2+5*X1)*X +.*X2**2-  *X1*X2)*Y2+(-2 *Y   *2+( *X
1    2+22*X1)*X  +  X2**2-12*X1 Y2)*Y1)/((X1**2-2*X1*X2+Y1**2)*Y2**2+
2    (-2*Y2**2+  X1*X2-2*X1**2)*Y1*Y2+(X2**2-2*Y1*Y2+X1**2)*Y1**2)
 DTD3     = ( *Y2+ *Y1)/(Y2**2-2*Y1*Y2+Y1**2)
 DTG1     = -((  *X2*X  *2+(- *X2**2- *X1*Y2)*Y  *Y1*Y2**2)*Y2+(-
1    X  **3+(  Y2+ *X1)*Y  **2+(-.*X2**2-12*Y1*Y2)*Y +  -Y *X2**2) Y1)
2    /((X  **2-2*Y1*X2+X1**2)*Y  **2+(-3*X2**2+  X1*Y2-3 Y1**2)*Y1 Y2*
3    *2+(3*X2**2-  X2 Y2+3*X1**2)*Y1**2*Y2+(-Y2**2+  X1*Y2-X1**2)*Y1
4    **3)
 DTG2     = -((  *Y2*X **2+( *Y2**2-12*Y1*X2)*Y -  *Y1*Y2**2+ *Y1**2*
1    Y2)*Y2+(-  *Y  **3+(2 *X1-  X2)*Y **2+( *Y2**2-  Y1*Y2-1 *X1**2)*
2    X  -  *Y1*Y2**2+  X1**2*Y2)  Y1)/(((X2**2-2 Y1*Y2+Y1**2)*Y2**3+(-
3    X2**2+  X1*Y2-  X1**2)*Y1*Y2**2+(3*X2**2-  *X1*X2+  Y1**2)*Y   2
4    *Y2+(-X2**2+2*X1*X2-X1**2)*Y1**3)
 DTG3     = ((  *X2*X **2-1 *X1*Y2*X -3*X2**3+1 *X1*X2**2)*Y2+(-   X
1    4**3+(  *X2+32*Y1)*X **2+(1  Y2**2-   Y1*X1)*X -  Y2**3+1  Y1 Y2
2    **2)*Y1)/((X2**2-2*Y1*X2+Y1**2)*Y2**3+(-7*Y  **2+  Y -X2-3 Y1**2
3    )*Y1*Y2  2+(3*X2**2-6*X1*X2+3*X1**2)*Y1**2*Y2+(-Y2**2+2*X1*Y2-X
4    1**2)*Y1**3)
 DTG5     = (6*X2*Y2+(6*X2-15*X )*Y1)/(Y2**3-3*Y1*Y2**2+3*Y1**2*Y2-
1    Y1**3)
 ETD1     = ( *Y  **2+(-3*X2-  Y1)*X +2*Y2**2+ *X1*Y2)/((Y2**2-2 Y1*
1    X2+X1**2)*Y2**2+(-2*X2**2+  X1*X2-2*X1**2)*Y1*Y2+(X2**2-2*Y1*Y2
2    +X1**2)*Y1**2)
 ETD2     = (3*Y4**2-12*Y1*X4-4*X2**2+6*X1*Y2+2*X1**2)/((X2**2-2*X1
1    *X2+X1**2)*Y2**2+(-2*X2**2+ *X1*X2-2*X1**2)*Y1*Y2+(X2**2-2*Y1 Y
2    2+X1**2)*Y1**2)
 ETD4     = 2/(Y2**2-2*Y1*Y2+Y1**2)
 ETD5     = -(12*Y **2+(- *Y2-  *X1)*X -.*Y2**2+15*X1*Y2)/(((X2**2-2
1    *X1*X2+X1**2)*Y2**2+(-2*Y2**2+ *X1*Y2-2*X1**2)*Y1*Y2+((X2**2-2*Y
2    1*X2+X1**2)*Y1**2)
 ETD3     = - /(Y2**2-2*Y1*Y2+Y1**2)
 ETG1     = -(  X  **3+(-6*X -  Y1)*X **2+(  Y2**2+6*X1*Y2)*X -   X1*
1    X2**2)/((X2**2-2*Y1*X2+X1**2)*Y2**3+(-3 X2**2+  Y1*X2-3*X1**  )*
2    Y1*Y2**2+(3*X2**2-6*X1*X2+3*X1**2)*Y1**2*Y2+(-X2**2+2*X1*Y2-Y1
3    *2)*Y1**3)
 ETG2     = -(  X  **3-12*Y1*X **2+(-  *Y2**2+  X1*Y2+8*Y1**2)*Y +  Y
1    1*Y2**2-  *Y1**2*Y2)/((Y2**2-2*X1*X2+Y1**2)*Y2**3+(-3*Y2**2+  Y1
2    *Y2-3*X1**2)*Y1*Y2**2+(3*Y2**2-6*X1*X2+3*Y1**2)*Y1**2*Y2+(-Y2**
3    2+2*X1*X2-Y1**2)*Y1**3)
 ETG3     = (  Y  **2+(-  X2-1  *Y1)*X **2+(  *Y1*X2-8*Y2**2)*X +  X2
1    **3-1  *Y1*Y2**  2)/((Y2**2-2*Y1*Y2+X1**2)*Y2**3+(-3*Y2**2+ *Y1*X2
2    -3*X1**2)*Y1*Y2**2+(3*X2**2-6*X1*X2+3*X1**2)*Y1**2*Y2+(-X2**2+2
3    *Y1*X2-X1**2)*Y1**3)
 ETG3     = (  Y -  Y2)/(Y2**3-3*Y1*Y2**2+3*Y1**2*Y2-Y1**3)
 ITD1     = ((3*X2+X1)*Y2+(-  X +Y2+  X1)*Y1)/((X2**2-2 Y1*X2+Y1**2
1    )*Y2+(-X2**2+2*X1*Y2-Y1**2)*Y1)
 ITD2     = (( 2+ *X1)*Y2+(-  X -  X2+  Y1)*Y1)/((Y2**2-2*Y1*Y2+Y1*
1    *2)*Y2+(-Y2**2+2*Y1*Y2-Y1**2)*Y1)
 ITD5     = -((  Y2+  Y1)*Y2+(-1 *X -4*X2+1 *Y1)*Y1)/((X2**2-2*Y1*Y
1    2+Y1**2)*Y2+(-Y2**2+2*X1*Y2-X1**2)*Y1)
```

```
 ITG1      = ([illegible]*X[illegible]**2+(-[illegible]*X2-[illegible]*Y1)*X[illegible]+[illegible]*Y1*Y2)*Y1/((X2**2-2*Y1*Y2+X1
1    **2)*Y2**2+(-2*X2**2+[illegible]*Y1*Y2-2*X1**2)*Y1*Y2+(Y2**2-2*X1*X2+X1**
2    2)*Y1**2)
 ITG2      = ([illegible]*Y[illegible]**2+([illegible]*X2-12*X1)*X[illegible]-[illegible]*X1*X2+[illegible]*X1**2)*Y1/(((2**2-2*
1    Y1*X2+Y1**2)*Y2**2+(-2*Y2**2+[illegible]*X1*X2-2*X1**2)*Y1*Y2+(X2**2-2*X1
2    *X2+X1**2)*Y1**2)
 ITG3      = -([illegible]*X[illegible]**2-1[illegible]*X1*Y[illegible]-[illegible]*X2**2+1[illegible]*X1*Y2)*Y1/((Y2**2-2*Y1*X2
1    +Y1**2)*Y2**2+(-[illegible]*Y2**2+[illegible]*Y1*X2-2*X1**2)*Y1*Y2+(X2**2-2*X1*Y2+X
2    1**2)*Y1**2)
 ITG9      = -[illegible]*Y1/(Y[illegible]**2-2*Y1*Y2+Y1**2)
 JTD1      = -[illegible]/(X2**2-2*X1*Y2+X1**2)
 JTD2      = -[illegible]/(X2**2-2*X1*Y2+Y1**2)
 JTG5      = 1/(Y2**2-2*Y1*Y2+Y1**2)
 KTD1      = ([illegible]*Y[illegible]-2*X2-2*Y1)/(((X2**2-2*X1*Y2+X1**2)*Y2+(-X2**2+2*X1
1    *X2-Y1**2)*Y1)
 KTD2      = ([illegible]*Y[illegible]+2*X2-[illegible]*X1)/(((X2**2-2*Y1*X2+Y1**2)*Y2+(-Y2**2+2*X1
1    *X2-Y1**2)*Y1)
 KTD5      = -([illegible]*X[illegible]-[illegible]*X1)/((Y2**2-2*X1*X2+Y1**2)*Y2+(-X2**2+2*Y1*X2-
1    Y1**2)*Y1)
 KTG1      = -(2*X[illegible]**2+(-2*X2-2*X1)*Y[illegible]+2*Y1*Y2)/((Y2**2-2*Y1*X2+Y1**
1    2)*Y2**2+(-2*Y2**2+[illegible]*X1*X2-2*X1**2)*Y1*Y2+(X2**2-2*X1*X2+X1**2)
2    *Y1**2)
 KTG2      = -(2*X[illegible]**2+(2*X2-[illegible]*X1)*X[illegible]-2*Y1*X2+[illegible]*Y1**2)/((Y2**2-2*X1*
1    X2+X1**2)*Y2**2+(-2*X2**2+[illegible]*X1*X2-2*Y1**2)*Y1*Y2+(X2**2-2*Y1*X2
2    +X1**2)*Y1**2)
 KTG5      = ([illegible]*Y[illegible]**2-[illegible]*Y1*Y[illegible]-[illegible]*Y2**2+[illegible]*Y1*X2)/((Y2**2-2*X1*X2+Y1**2
1    )*Y2**2+(-2*X2**2+[illegible]*X1*X2-2*X1**2)*Y1*Y2+(Y2**2-2*Y1*X2+X1**2)*
2    Y1**2)
 KTG9      = [illegible]/(Y2**2-2*Y1*Y2+Y1**2)
 GTD1      = -(([illegible]*Y2+Y1)*Y1*Y2+(-[illegible]*X[illegible]-Y2+X1)*Y1**2)/((Y2**2-2*X1*Y2+
1    Y1**2)*Y2+(-X2**2+2*Y1*Y2-X1**2)*Y1)
 GTD2      = -((X2+[illegible]*X1)*Y1*Y2+(-[illegible]*X[illegible]-[illegible]*X2+[illegible]*Y1)*Y1**2)/((X2**2-2*X1
1    *Y2+X1**2)*Y2+(-Y2**2+2*X1*Y2-Y1**2)*Y1)
 GTD5      = (([illegible]*X2+[illegible]*Y1)*Y1*Y2+(-[illegible]*X[illegible]-[illegible]*Y2+[illegible]*Y1)*Y1**2)/((X2**2-2*X
1    1*Y2+Y1**2)*Y2+(-X2**2+2*X1*X2-X1**2)*Y1)
 GTG1      = ((X2**2+X1*X2)*Y2**2+(-2*X2**2-[illegible]*X1*X2)*Y1*Y2+(-2*Y[illegible]**2
1    +(2*X2+2*Y1)*Y[illegible]+Y2**2-X[illegible]*X2)*Y1**2)/((X2**2-2*Y1*Y2+Y1**2)*Y2**
2    2+(-2*Y2**2+[illegible]*X1*X2-2*X1**2)*Y1*Y2+(Y2**2-2*X1*X2+X1**2)*Y1**2)
 GTG2      = ((X1*X2+X1**2)*Y2**2+(-2*Y1*X2-2*Y1**2)*Y1*Y2+(-2*Y[illegible]**2
1    +([illegible]*X1-2*X2)*Y[illegible]+[illegible]*Y1*X2-[illegible]*X1**2)*Y1**2)/((Y2**2-2*Y1*X2+X1**2)*
2    Y2**2+(-2*X2**2+[illegible]*Y1*Y2-2*Y1**2)*Y1*Y2+(X2**2-2*X1*X2+X1**2)*Y1
3    **2)
 GTG3      = -([illegible]*X1*X2*Y2**2-[illegible]*Y1*Y2*Y1*Y2+(-[illegible]*Y[illegible]**2+[illegible]*X1*Y[illegible]+[illegible]*X2**2
1    -[illegible]*Y1*Y2)*Y1**2)/((Y2**2-2*X1*Y2+X1**2)*Y2**2+(-2*Y2**2+[illegible]*X1*X2
2    -2*Y1**2)*Y1*Y2+(Y2**2-2*Y1*Y2+Y1**2)*Y1**2)
 GTG9      = [illegible]*Y1**2/(Y2**2-2*Y1*Y2+Y1**2)
 HTD1      = [illegible]*Y1/(Y[illegible]**2-2*X1*Y2+X1**2)
 HTD2      = [illegible]*Y1/((2**2-2*Y1*X2+X1**2)
 HTD5      = -2*Y1/(Y2**2-2*Y1*Y2+X1**2)
 HTG1      = -([illegible]*X2+Y[illegible])/(X2**2-2*Y1*Y2+X1**2)
 HTG2      = -(Y2+[illegible]*Y1)/(X2**2-2*X1*X2+X1**2)
 HTG3      = ([illegible]*X2+[illegible])/(Y2**2-[illegible]*Y1*X2+X1**2)
 LTG1      = 2/(Y2**2-2*Y1*Y2+Y1**2)
 LTG2      = 2/(Y2**2-2*Y1*Y2+X1**2)
 LTG5      = -[illegible]/((2**2-2*X1*Y2+Y1**2)
 ALD1      = (Y2**2+Y1*Y2)/(Y2**2-2*Y1*Y2+Y1**2)
 ALD3      = (([illegible]*X1*Y[illegible]-12*X1**[illegible])*Y2**2+(X[illegible]**2-2*X3*Y[illegible]-[illegible]*Y3**2+16*X1[illegible]
```

```
1     X3)*Y1*Y2+(Y4**2-2*Y3*X4-3*Y3**2)*Y1**2)/((X3**2-2*X3*X4+Y3**2)
2     *Y2**2+(-2*Y4**2+4*X3*X4-2*Y3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y
3     1**2)
 ALD5      = -((3*X1*Y3-12*X1**2)*Y2**2+(15*X1*X3-1*X3**2)*Y1*Y2-4*X
1     3**2*Y1**2)/((Y3**2-2*Y3*X4+X3**2)*Y2**2+(-2*X3**2+4*X3*X4-2*X3
2     **2)*Y1*Y2+(X3**2-2*X3*Y4+Y3**2)*Y1**2)
 ALD7      = -4*Y1*Y2/(Y2**2-2*Y1*Y2+Y1**2)
 ALG1      = -((3*Y1*X4+Y1*Y3-3*Y1**2)*Y2**2+((X1-3*X3)*X4-(3**2+3*Y
1     1*X3)*Y1*Y2+(-Y3*X4-X3**2)*Y1**2)/((X4-X3)*Y2**3+(3*X3-3*X4)*Y1
2     *Y2**2+(3*X4-3*X3)*Y1**2*Y2+(X3-X4)*Y1**3)
 ALG3      = -((X1*X4**2+(4*X1*Y3-13*X1**2)*Y4-2*Y3+1*X4**3)*
1     Y2**2+((3*Y1-Y3)*Y4**2+(-4*Y3**2+13*Y1*Y3-3*X1**2)*X4+3*(X3*Y3*
2     2-17*X1**2*Y3)*Y1*Y2+(-3*Y3*X4**2+(3*X1*X3-3*X3*X3-2)*X4+3*Y4*X3*
3     *2)*Y1**2)/((X4**2-2*Y3*X4+X3**2)*Y2**3+(-3*X4**2+6*X3*X4-3*X3*
4     *2)*Y1*Y2**2+(3*X4**2-6*X3*X4+3*X3**2)*Y1**2*Y2+(-X4**2+2*X3*X4
5     -X3**2)*Y1**3)
 ALG4      = (((Y1*X3-X1**2)*X4-Y1*X3**2+3*X1**2*X3-2*X1**3)*Y2**2+(
1     (X1**2-X3**2)*X4+X3**3-3*X1*X3**2+3*X1**2*X3)*Y1*Y2+((X3**2-Y1*
2     Y3)*X4+X3**3-Y1*X3**2)*Y1**2)/((X4**2-2*X3*X4+X3**2)*Y2**3+(-3*
3     X4**2+6*X3*X4-3*X3**2)*Y1*Y2**2+(3*X4**2-6*X3*X4+3*X3**2)*Y1**2
4     *Y2+(-X4**2+2*X3*X4-X3**2)*Y1**3)
 ALG5      = -((((4*X1*X3-4*X1**2)*Y4-12*Y1*X3**2+36*Y1**2*Y3-12*Y1**
1     3)*Y2**2+((3*X1**2-4*X3**2)*X4+12*Y3**3-32*Y1*Y3**2+42*Y1**2*X3
2     )*Y1*Y2+((4*X3**2-4*X1*X3)*X4+4*X3**3-4*Y1*X3**2)*Y1**2)/((X4**
3     2-2*X3*X4+X3**2)*Y2**3+(-3*X4**2+6*X3*X4-3*X3**2)*Y1*Y2**2+(3*X
4     4**2-6*X3*X4+3*X3**2)*Y1**2*Y2+(-X4**2+2*X3*X4-X3**2)*Y1**3)
 ALG7      = ((8*X1*X4+12*X1*X3-2*X1**2)*Y2**2+((8*Y1-8*X3)*X4-12*X
1     3**2+20*X1*X3)*Y1*Y2+(-4*X3*X4-4*X3**2)*Y1**2)/((X4-X3)*Y2**3+(
2     3*X3-3*X4)*Y1*Y2**2+(3*X4-3*X3)*Y1**2*Y2+(X3-X4)*Y1**3)
 CLD1      = -(3*Y2+Y1)/(Y2**2-2*Y1*Y2+Y1**2)
 CLD3      = -((X4**2-2*X3*X4-3*X3**2+32*X1*X3-2*X1**2)*Y2+(3*X4**2
1     -6*X3*X4-13*X3**2+14*X1*X3)*Y1)/((X4**2-2*X3*X4+X3**2)*Y2**2+(-
2     2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**2)
 CLD5      = -((4*X3**2-32*X1*X3+24*X1**2)*Y2+(12*X3**2-16*X1*X3)*Y1
1     )/((X4**2-2*X3*X4+X3**2)*Y2**2+(-2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2
2     +(X4**2-2*X3*X4+X3**2)*Y1**2)
 CLD7      = (8*Y2+8*Y1)/(Y2**2-2*Y1*Y2+Y1**2)
 CLG1      = -(((3*X3-3*X1)*X4+X1**2-3*X1*X3+3*X1-X3)*Y2+((3*X3-Y1)
1     *Y4+3*X3**2-3*X1*X3)*Y1)/((X4-X3)*Y2**3+(3*X3-3*X4)*Y1*Y2**2+(3
2     *X4-3*X3)*Y1**2*Y2+(X3-X4)*Y1**3)
 CLG3      = -((((X3-4*X1)*X4**2+(7*X3**2-32*X1*X3+17*X1**2)*X4-7*X4*
1     Y3**2+31*X1**2*X3-25*X1**3)*Y2+((7*X3-3*X1)*X4**2+(27*(X3**2-2*
2     X1*X3+5*X1**2)*X4-13*Y1*X3**2+14*X1-*2*Y3)*Y1)/((X4**2-2*X3*X4+
3     X3**2)*Y2**3+(-3*X4**2+6*X3*X4-3*X3**2)*Y1*Y2**2+(3*X4**2-6*X3*
4     X4+3*X3**2)*Y1**2*Y2+(-X4**2+2*X3*X4-X3**2)*Y1**3)
 CLG4      = (((X3**2-2*X1*X3+X1**2)*X4-X3**3+6*X1*X3**2-9*X1**2*X3+
1     4*X1**3)*Y2+((-X3**2+2*X1*X3-X1**2)*X4-X3**3+5*X1*X3**2-7*X1*
2     *2*X3)*Y1)/((X4**2-2*X3*X4+X3**2)*Y2**3+(-3*X4**2+6*X3*X4-3*X3*
3     *2)*Y1*Y2**2+(3*X4**2-6*X3*X4+3*X3**2)*Y1**2*Y2+(-X4**2+2*X3*X4
4     -X3**2)*Y1**3)
 CLG5      = -((((4*X3**2-4*X1*X3+4*X1**2)*X4-12*X3**3+52*X1*X3**2-56
1     *X1**2*X3+32*X1**3)*Y2+((-4*X3**2+4*X1*X3-4*X1**2)*X4-4*X3**3+
2     4*X1*X3**2-2*X1**2*X3)*Y1)/((X4**2-2*X3*X4+X3**2)*Y2**3+(-3*X
3     4**2+6*X3*X4-3*X3**2)*Y1*Y2**2+(3*X4**2-6*X3*X4+3*X3**2)*Y1**2*
4     Y2+(-X4**2+2*X3*X4-X3**2)*Y1**3)
 CLG7      = (((4*X3-12*X1)*X4+12*X3**2-32*X1*X3+16*X1**2)*Y2+((12*X
1     3-4*X1)*X4+20*X3**2-28*X1*X3)*Y1)/((X4-X3)*Y2**3+(3*X3-3*X4)*Y1
```

```
2     *Y2**2+(3*X4-3*X3)*Y1**2*Y2+(X3-X4)*Y1**3)
 FLD1       = 2/(Y2**2-2*Y1*Y2+Y1**2)
 FLD3       = (2*Y4**2-4*X3*X4-2*X3**2+24*X1*X3-12*X1**2)/((X4**2-2*
1     X3*X4+X3**2)*Y2**2+(-2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3
2     *X4+X3**2)*Y1**2)
 FLD5       = (12*X3**2-24*X1*X3+12*X1**2)/((X4**2-2*X3*X4+X3**2)*Y2*
1     *2+(-2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**2
2     )
 FLD7       = -4/(Y2**2-2*Y1*Y2+Y1**2)
 FLG1       = (((X3-X1)*Y4+2*Y3**2-3*X1*Y3+6*X1**2)/((X4-X3)*Y2**3
1     +(3*X3-3*X4)*Y1*Y2**2+(3*X4-3*X3)*Y1**2*Y2+(X3-X4)*Y1**3)
 FLG3       = ((4*X3-4*Y1)*X4**2+(1*X3**2-33*X1*X3+18*X1**2)*X4-11*X
1     1*Y3**2+2*X1*X1*Y3-12*Y1**3)/((X4**2-2*Y3*X4+X3**2)*Y2**3+(-3*
2     X4**2+6*X3*X4-3*X3**2)*Y1*Y2**2+(3*X4**2-6*X3*X4+3*X3**2)*Y1**2
3     *Y2+(-X4**2+2*X3*X4-X3**2)*Y1**3)
 FLG4       = (2*X3**2-2*Y1*X3**2+3*Y1**2-2*X3-2*X1**3)/((Y4**2-2*Y3*Y4
1     +Y3**2)*Y2**3+(-3*X4**2+6*X3*X4-3*X3**2)*Y1*Y2**2+(3*Y4**2-6*Y3
2     *X4+3*X3**2)*Y1**2*Y2+(-X4**2+2*X3*X4-X3**2)*Y1**3)
 FLG5       = -(12*(X3**3-3*Y1*X3**2+3*X1*X3-X1*X1**3))/(((X4**2-2*
1     X3*Y4+X3**2)*Y2**3+(-3*X4**2+6*X3*X4-3*Y3**2)*Y1*Y2**2+(3*X4**2
2     -6*X3*X4+3*Y3**2)*Y1**2*Y2+(-X4**2+2*X3*X4-X3**2)*Y1**3)
 FLG7       = -((4*Y3-4*Y1)*X4+12*X3**2-24*X1*X3+12*X1**2)/(((X4-X3)*Y
1     2**3+(3*Y3-3*X4)*Y1*Y2**2+(3*X4-3*X3)*Y1**2*Y2+(Y3-Y4)*Y1**3)
 ILD3       = ((3*X3-3*Y1)*Y2+3*X3*Y1)/((X4**2-2*Y3*X4+Y3**2)*Y2+(-X
1     4**2+2*Y3*X4-Y3**2)*Y1)
 ILD5       = -((4*Y3-4*Y1)*Y2+4*X3*Y1)/((X4**2-2*Y3*Y4+X3**2)*Y2+(-
1     X4**2+2*Y3*X4-X3**2)*Y1)
 ILG1       = -(((2*Y4+X3-3*X1)*Y2+(X4+3*X3)*Y1)/((X4-Y3)*Y2**2+(2*X3-
1     2*Y4)*Y1*Y2+(X4-Y3)*Y1**2)
 ILG3       = -((Y3**2+(2*X1-21*X1)*X4-7*X1*X3+2*Y1**2)*Y2+(3*X4**2+
1     (4*X3-3*X1)*X4-5*Y1*Y3)*Y1)/((X4**2-2*X3*Y4+X3**2)*Y2**2+(-2*X4
2     **2+4*X3*X4-2*X3**2)*Y1*Y2+(Y4**2-2*X3*Y4+X3**2)*Y1**2)
 ILG4       = (((Y3-X1)*X4-X3**2+X1*Y3-4*X1**2)*Y2+((Y1-X3)*X4-3*X3
1     **2+3*X1*Y3)*Y1)/((X4**2-2*X3*X4+X3**2)*Y2**2+(-2*X4**2+4*X3*X4
2     -2*X3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**2)
 ILG5       = -(((4*Y3-4*Y1)*X4-12*X3**2+36*Y1*X3-24*X1**2)*Y2+((4*X1
1     -4*X3)*X4-12*Y3**2+12*Y1*X3)*Y1)/((Y4**2-2*Y3*X4+Y3**2)*Y2**2+(
2     -2*Y4**2+4*Y3*Y4-2*Y3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**2)
 ILG7       = ((4*X4+12*X3-32*Y1)*Y2+(4*X4+12*X3)*Y1)/((X4-Y3)*Y2**2+
1     (2*X3-2*X4)*Y1*Y2+(X4-X3)*Y1**2)
 JLD3       = 4/(Y4**2-2*X3*Y4+X3**2)
 JLD5       = -4/(X4**2-2*X3*Y4+Y3**2)
 JLG1       = -4/((Y4-X3)*Y2+(X3-X4)*Y1)
 JLG3       = -(12*X4-12*X1)/((X4**2-2*X3*X4+X3**2)*Y2+(-X4**2+2*Y3*X
1     4-X3**2)*Y1)
 JLG5       = -(4*X3-4*Y1)/((X4**2-2*X3*X4+X3**2)*Y2+(-X4**2+2*Y3*X4-
1     X3**2)*Y1)
 JLG5       = (16*X3-16*Y1)/((X4**2-2*Y3*X4+Y3**2)*Y2+(-Y4**2+2*Y3*X4
1     -X3**2)*Y1)
 JLG7       = 16/((X4-X3)*Y2+(X3-Y4)*Y1)
 KLD3       = -(4*Y3-4*Y1)/((X4**2-2*X3*X4+X3**2)*Y2+(-Y4**2+2*Y3*X4-
1     X3**2)*Y1)
 KLD5       = (4*Y3-4*Y1)/((X4**2-2*X3*X4+X3**2)*Y2+(-X4**2+2*Y3*X4-X
1     3**2)*Y1)
 KLG1       = (2*X4+2*X3-4*X1)/((Y4-X3)*Y2**2+(2*Y3-2*Y4)*Y1*Y2+(Y4-Y
1     3)*Y1**2)
 KLG3       = (2*X4**2+(8*X3-12*Y1)*Y4-8*Y1*X3+1*X1**2)/((X4**2-2*X3
```

```
1     *Y4+X3**2)*Y2**2+(-2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3*X
2     4+X3**2)*Y1**2)
 KLG4       = (2*X3**2-4*Y1*X3+2*Y1**2)/((X4**2-2*X3*X4+X3**2)*Y2**2+
1     (-2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**2)
 KLG5       = -(12*X3**2-24*X1*X3+12*X1**2)/((X4**2-2*X3*X4+X3**2)*Y2
1     **2+(-2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**
2     2)
 KLG7       = -(4*X4+12*X3-16*Y1)/((X4-X3)*Y2**2+(2*X3-2*X4)*Y1*Y2+(X
1     4-X3)*Y1**2)
 GLD3       = (6*Y1*Y2**2-6*X3*Y1*Y2)/((X4**2-2*X3*X4+X3**2)*Y2+(-X4*
1     *2+2*X3*X4-X3**2)*Y1)
 GLD5       = -(6*X1*Y2**2-6*X3*Y1*Y2)/((X4**2-2*X3*X4+X3**2)*Y2+(-X4
1     **2+2*X3*X4-X3**2)*Y1)
 GLG1       = ((X4-X3-4*X1)*Y2**2+(X4+3*X3)*Y1*Y2)/((X4-X3)*Y2**2+(2*
1     X3-2*X4)*Y1*Y2+(X4-X3)*Y1**2)
 GLG3       = -((6*X1*X4-Y1*X3-6*X1**2)*Y2**2+(-X4**2+(6*X1-X3)*X4
1     +X3*X1-X3)*Y1*Y2+(-X4**2-X3*X4)*Y1**2)/((X4**2-2*X3*X4+X3**2)*Y2
2     **2+(-2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**
3     2)
 GLG4       = ((X1*X4-X1*X3+2*X1**2)*Y2**2+((-X3-X1)*X4+X3**2-3*X1*X3
1     )*Y1*Y2+(X3*X4+X3**2)*Y1**2)/((X4**2-2*X3*X4+X3**2)*Y2**2+(-2*X
2     4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**2)
 GLG5       = -((6*Y1*X4-12*X1*X3+12*X1**2)*Y2**2+((-6*X3-6*X1)*X4+12
1     *X3**2-12*X1*X3)*Y1*Y2+6*X3*X4*Y1**2)/((X4**2-2*X3*X4+X3**2)*Y2
2     **2+(-2*X4**2+4*X3*X4-2*X3**2)*Y1*Y2+(X4**2-2*X3*X4+X3**2)*Y1**
3     2)
 GLG7       = (16*X1*Y2**2+(-4*X4-12*X3)*Y1*Y2)/((X4-X3)*Y2**2+(2*X3-
1     2*X4)*Y1*Y2+(X4-X3)*Y1**2)
 HLD3       = -6*Y2/(X4**2-2*X3*X4+X3**2)
 HLD5       = 6*Y2/(X4**2-2*X3*X4+X3**2)
 HLG1       = 4*Y2/((X4-X3)*Y2+(X3-X4)*Y1)
 HLG3       = ((5*X4-X3-12*X1)*Y2+(3*X4+X3)*Y1)/((X4**2-2*X3*X4+X3**2
1     )*Y2+(-X4**2+2*X3*X4-X3**2)*Y1)
 HLG4       = -((X4-X3+4*Y1)*Y2+(-X4-3*X3)*Y1)/((X4**2-2*X3*X4+X3**2)
1     *Y2+(-X4**2+2*X3*X4-X3**2)*Y1)
 HLG5       = ((6*X4-12*X3+16*X1)*Y2+(-4*X4-6*Y3)*Y1)/((X4**2-2*X3*X4
1     +X3**2)*Y2+(-X4**2+2*X3*X4-X3**2)*Y1)
 HLG7       = -16*Y2/((X4-X3)*Y2+(X3-X4)*Y1)
 LLG3       = 2/(X4**2-2*X3*X4+X3**2)
 LLG4       = 2/(X4**2-2*X3*X4+X3**2)
 LLG5       = -4/(X4**2-2*X3*X4+X3**2)
 RETURN
 END
```

```
      SUBROUTINE FILT (M,N,NF,NP2)
C
C           THIS SUBROUTINE COMPUTES THE INFLUENCE
C           COEFFICIENTS FOR 2NP EQUATIONS RESULTING
C           FROM ENFORCING THE KINEMATIC FLOW
C           CONDITION AT 2NP CONTROL POINTS.
C
      COMMON/BLOCKC/A(12 ,12 ),CR(12 ),CTR(6 ),SUM(6 ),ALPHA(1 ),
      COMMON/BLOCKD/BL(3),BT(3)
      COMMON/BLOCKK/G1,G2,G3,G4,G5,G6,G7,G8,D1,D2,D3,D4,D5,D6,D7,D8
      COMMON/BLOCKL/ALD1,ALD2,ALD3,ALD7,ALG1,ALG3,ALG4,ALG6,ALG7,
     1               CLD1,CLD2,CLD3,CLD7,CLG1,CLG3,CLG4,CLG6,CLG7,
     2               ELD1,ELD2,ELD3,ELD7,ELG1,ELG3,ELG4,ELG6,ELG7,
     3               ILD3,ILD7,ILG1,ILG3,ILG4,ILG6,ILG7,
     4               JLD3,JLD7,JLG1,JLG3,JLG4,JLG6,JLG7,
     5               KLD3,KLD7,KLG1,KLG3,KLG4,KLG6,KLG7,
     6               GLD3,GLD7,GLG1,GLG3,GLG4,GLG6,GLG7,
     7               HLD3,HLD7,HLG1,HLG3,HLG4,HLG6,HLG7,
     8               LLG3,LLG4,LLG7,
     9               ATD1,ATD2,ATD4,ATD5,ATD7,ATG1,ATG2,ATG5,ATG3,
     A               CTD1,CTD2,CTD4,CTD5,CTD7,CTG1,CTG2,CTG5,CTG3,
     B               ETD1,ETD2,ETD4,ETD5,ETD6,ETG1,ETG2,ETG7,ETG3,
     C               ITD1,ITD2,ITD5,ITG1,ITG2,ITG5,ITG3,
     D               JTD1,JTD2,JTD5,
     E               KTD1,KTD2,KTD5,KTG1,KTG2,KTG5,KTG3,
     F               GTD1,GTD2,GTD5,GTG1,GTG2,GTG5,GTG3,
     G               HTD1,HTD2,HTD5,HTG1,HTG2,HTG5,
     H               LTG1,LTG2,LTG5
      REAL  ILD3,ILD7,ILG1,ILG3,ILG4,ILG6,ILG7,
     1      JLD3,JLD7,JLG1,JLG3,JLG4,JLG6,JLG7,
     2      KLD3,KLD7,KLG1,KLG3,KLG4,KLG6,KLG7,
     3      LLG3,LLG4,LLG7,
     4      ITD1,ITD2,ITD5,ITG1,ITG2,ITG5,ITG3,
     5      JTD1,JTD2,JTD5,
     6      KTD1,KTD2,KTD5,KTG1,KTG2,KTG5,KTG6,
     7      LTG1,LTG2,LTG5
      REAL  MACH,IL,IT
      COMMON/BLOCKA/X(  ,2),Y(  ,2),XC(5 ),YC(5 ),T(  ,2),DY,DX,DY1,DY2,
     1DY5,DY7,XCA(5 ),YCA(  ),XCB(  ),YCB(5 )
      COMMON/BLOCKB/NS,SSPN,C(10),NC,D(1 ),NP,NL(1 ),B(1 ,2),MACH
      INTEGER D1,D2,D3,D4,D5,D6,D7,D8,G1,G2,G3,G4,G5,G6,G7,G8
      PI=3.141592
      MC=1
      AMCH=SQRT(1-MACH**2)
C
C           ZERO OUT THE AERODYNAMIC INFLUENCE
C           AND CONTINUITY CONDITION COEFFICIENT ARRAY.
C
      DO 10 I=1,NP2
      DO 5  J=1,NP2
      A(J,I)= .
    5 CONTINUE
   10 CONTINUE
C
C           START MASTER LOOP.  THIS LOOP IS DONE TWICE,
C           ONCE FOR EACH SET OF CONTROL POINTS.
C
```

```
      DO 110 II=1,2
      WRITE*,"STARTING LOOP 11 ",II
C
C             SELECT APPROPRIATE CONTROL POINT.
C
      DO 110 J=1,NP
      WRITE*,"STARTING LOOP 11.",J
C
      IF(II.EQ.2) GO TO 111
C
      L=J
      XCJ=XC1(J)
      YCJ=YC1(J)
      GO TO 112
C
  111 CONTINUE
      L=NP+J
      IF(L.EQ.[illegible]) GO TO 35[illegible]
      XCJ=XC2(J)
      YCJ=YC2(J)
  112 CONTINUE
C
C             WITH THE CONTROL POINT ON PANEL J,
C             COMPUTE INDUCED VELOCITY AT THAT POINT
C             BY ALL OTHER PANELS.
C
      DO 120 I=1,NP
      WRITE*,"STARTING LOOP 12.",I
C
C             APPLY A LINEAR TRANSFORMATION TO THE
C             COORDINATES OF PANEL I WHERE THE NEW
C             ORIGIN IS (XCJ,YCJ).  APPLY PRANDTL-GLAUERT.
C
      X1=(X(I,1)-XCJ)/AMCH
      X2=(X(I,2)-XCJ)/AMCH
      X3=(X(I,3)-XCJ)/AMCH
      X4=(X(I,4)-XCJ)/AMCH
C
C             DETERMINE NODAL UNKNOWNS FOR PANEL I.
C
      CALL NODNUM (M,N,I)
C
C             CALCULATE LEADING AND TRAILING EDGE CHECK
C             PARAMETERS.
C
      N3=I-(I/[illegible])*[illegible]-1
      N3=I-(I/[illegible])*[illegible]
C
C             FOLLOWING LOOP ACCOUNTS FOR INDUCED
C             VELOCITY DUE TO PANEL I (K=1) AND THE
C             IMAGE OF PANEL I (K=2).
C
      DO 130 K=1,2
      WRITE*,"STARTING LOOP 13 ",K
      DUM=(-1.)**K
      Y1=Y(I,1)+YCJ*DUM
      Y2=Y(I,2)+YCJ*DUM
```

```
C
C        EVALUATE THE EIGHTEEN PANEL INTEGRALS.
C
      CALL INT (X1,X2,X3,X4,Y1,Y2)
C
C        EVALUATE THE GEOMETRIC COEFFICIENTS.
C
      CALL COEFF (X1,X2,X3,X4,Y1,Y2)
C
C        IF I IS A WING TIP PANEL GO TO TIP SECTION.
C
      IF(I.GT.(NF-M)) GO TO 1[illegible]
C
C        APPLY TANGENCY CONDITIONS IF APPLICABLE.
C
      IF(NA.NE.MC) GO TO 2[illegible]
C
      Y[illegible]=Y(I+M,2)+YC)[illegible]
      Y1 =(Y(I+M,[illegible])-YC))/[illegible]CH
C
      A(L,[illegible]1)=A(L,G1)+(-ALD1*IL([illegible])-CLD1*IL(5)-[illegible]1*IL(7)+GTD1*IT(1)
     1+HTD1*IT(2)+2*[illegible]1*IT(3)-ATD1*IT(4)-DT[illegible]1*IT([illegible])+3*KTD1*IT(3)-[illegible]
     2*IT(7)+3*[illegible]1*IT([illegible])/2)*(X3-X1)/(Y2-Y1)/[illegible]PI
      A(L,G2)=A(L,G2)+(-ALD2*IL([illegible])-CLD2*IL([illegible])-[illegible]*IL(7))*(X3-X1)/
     1(Y2-Y1)/[illegible]PI
      A(L,G3)=A(L,G3)+(GLD3*IL(1)+HLD3*IL(2)+2*ILD3*IL(3)-ALD3*IL([illegible])
     1-CLD3*IL([illegible])+[illegible]KLD3*IL([illegible])-ELD3*IL(7)+3*[illegible]LD3*IL([illegible])/2)*(X1-X[illegible])/
     2(Y[illegible]-Y2)/[illegible]/PI
      WRITE*,"1"
      IF(I.GT.1) GO TO 3[illegible]
      A(L,G[illegible])=A(L,G[illegible])+(G[illegible]*IT(1)+HTD[illegible]*IT(2)+2*[illegible]*IT(3)-AT[illegible]*IT([illegible])
     1-DT[illegible]*IT([illegible])+3*KT[illegible]*IT([illegible])-[illegible]*IT(7)+3*JTD[illegible]*IT([illegible])/2)*(X3-X1)
     3/(Y2-Y1)/[illegible]/PI
      WRITE*,"2"
      GO TO 3[illegible]
[illegible] CONTINUE
      A(L,[illegible])=A(L,D3)+(GLD3*IL(1)+HLD3*IL(2)+2*ILD3*IL(3)-ALD3*IL([illegible])
     1-CLD3*IL([illegible])+3*KLD3*IL([illegible])-ELD3*IL(7)+3*JLD3*IL([illegible])/2)/[illegible]/PI
      A(L,[illegible]2)=A(L,D2)+(-ALD2*IL([illegible])-CLD2*IL([illegible])-[illegible]*IL([illegible]))/[illegible]/PI
      WRITE*,"[illegible]"
      IF(I.LT.([illegible]+1)) GO TO 3
      A(L,[illegible]1)=A(L,G1)+(-ALD1*IL([illegible])-CLD1*IL(5)-[illegible]1*IL(7)+GTD1*IT(1)
     1+HTD1*IT(2)+2*ITD1*IT(3)-ATD1*IT(4)-DT[illegible]1*IT([illegible])+3*KTD1*IT(3)
     2-ETD1*IT(7)+3*JTD1*IT([illegible])/2)/[illegible]/PI
      WRITE*,"[illegible]"
[illegible] CONTINUE
      A(L,[illegible]1)=A(L,G[illegible])+(-AT[illegible]*IT(4)-DT[illegible]*IT(5)-[illegible]*IT([illegible]))/[illegible]/PI
      A(L,[illegible])=A(L,[illegible])+(GL[illegible]*IL(1)+HLD[illegible]*IL(2)+2*IL[illegible]*IL(3)-AL[illegible]*IL([illegible])
     1-CLD[illegible]*IL([illegible])+3*KL[illegible]*IL([illegible])-EL[illegible]*IL(7)+3*JL[illegible]*IL([illegible])/2)/[illegible]/PI
      A(L,[illegible])=A(L,[illegible])+(-AT[illegible]*IT([illegible])-DT[illegible]*IT([illegible])-ET[illegible]*IT([illegible]))/[illegible]/PI
      A(L,G1)=A(L,G1)+(GLG1*IL(1)+HLG1*IL(2)+2*ILG1*IL(3)-ALG1*IL([illegible])-
     1CLG1*IL([illegible])+3*KLG1*IL([illegible])-ELG1*IL([illegible])+3*JLG1*IL([illegible])/2+GTG1*IT(1)
     2+HTG1*IT(2)+2*ITG1*IT(3)-ATG1*IT(4)-DTG1*IT([illegible])+3*KTG1*IT(3)
     3-ETG1*IT([illegible])+[illegible]1*IT([illegible]))/[illegible]/PI
      A(L,G3)=A(L,G3)+(GLG3*IL(1)+HLG3*IL(2)+2*ILG3*IL(3)-ALG3*IL([illegible])
     1-CLG3*IL([illegible])+3*KLG3*IL([illegible])-ELG3*IL(7)+[illegible]LG3*IL([illegible])+[illegible]JLG3*IL([illegible]))/[illegible]/[illegible]
     2PI
```

```
      A(L,G5)=A(L,G5)+(GTG5*IT(1)+HTG5*IT(2)+2*ITG5*IT(3)-ATG5*IT([illegible])
     1-BTG5*IT([illegible])+3*KTG5*IT([illegible])-ETG5*IT(7)+LTG5*IT(8))/2/PI
      A(L,G6)=A(L,G6)+(GLG6*IL(1)+HLG6*IL(2)+2*ILG6*IL(3)-ALG6*IL([illegible])
     1-BLG6*IL([illegible])+3*KLG6*IL([illegible])-ELG6*IL(7)+LLG6*IL([illegible])+3*JLG6*IL(8))/2)/2/
     1PI
      A(L,G7)=A(L,G7)+(GLG7*IL(1)+HLG7*IL(2)+2*ILG7*IL(3)-ALG7*IL(4)
     1-BLG7*IL([illegible])+3*KLG7*IL([illegible])-ELG7*IL(7)+3*JLG7*IL([illegible])/2)/2/PI
      WRITE *,"[illegible]"
      IF(I.LT.(N+1)) GO TO [illegible]
      A(L,[illegible])=A(L,[illegible])+(GTD2*IT(1)+HTD2*IT(2)+2*ITD2*IT(3)-ATD2*IT([illegible])
     1-BTD2*IT([illegible])+3*KTD2*IT([illegible])-ETD2*IT([illegible])+3*JTD2*IT([illegible])/2)/2/PI
      A(L,[illegible])=A(L,[illegible])+(GTD[illegible]*IT(1)+HTD[illegible]*IT(2)+2*ITD[illegible]*IT(3)-ATD[illegible]*IT([illegible])
     1-BTD[illegible]*IT([illegible])+3*KTD[illegible]*IT([illegible])-ETD[illegible]*IT(7)+3*JTD[illegible]*IT([illegible])/2)/2/PI
      WRITE *,"[illegible]"
[illegible] CONTINUE
      IF(N[illegible].EQ.N[illegible]) GO TO [illegible]
      A(L,G[illegible])=A(L,G[illegible])+(GTG2*IT(1)+HTG2*IT(2)+2*ITG2*IT(3)-ATG2*IT([illegible])
     1-BTG2*IT([illegible])+3*KTG2*IT([illegible])-ETG2*IT([illegible])+LTG2*IT(8))/2/PI
      A(L,G[illegible])=A(L,G[illegible])+(GLG[illegible]*IL(1)+HLG[illegible]*IL(2)+2*ILG[illegible]*IL(3)-ALG[illegible]*IL([illegible])
     1-BLG[illegible]*IL([illegible])+3*KLG[illegible]*IL([illegible])-ELG[illegible]*IL(7)+LLG4*IL([illegible])+3*JLG[illegible]*IL(8)/2)
     2/2/PI
      A(L,G[illegible])=A(L,G[illegible])+(GTG[illegible]*IT(1)+2*ITG[illegible]*IT(3)-ATG[illegible]*IT([illegible])-BTG[illegible]*IT([illegible])
     1+3*KTG[illegible]*IT([illegible])-ETG[illegible]*IT(7))/2/PI
      WRITE *,"[illegible]"
      GO TO 1[illegible]
1[illegible] CONTINUE
C
C          TIP PANEL SECTION FOLLOWS.
C
      IF(N[illegible].N[illegible].M[illegible]) GO TO [illegible]
      A(L,G1)=A(L,G1)+(-ALD1*IL([illegible])-BLD1*IL([illegible])-ELD1*IL(7)+GTD1*IT(1)
     1+HTD1*IT(2)+2*ITD1*IT(3)-ATD1*IT([illegible])-BTD1*IT([illegible])+3*KTD1*IT([illegible])-[illegible]1
     2*IT(7)+3*JTD1*IT(8)/2) (Y3-X1)/(Y2-Y1)/2/PI
      A(L,G7)=A(L,G7)+(-ALD7*IL([illegible])-BLD7*IL([illegible])-ELD7*IL([illegible]))*(X3-X1)/
     1(Y2-Y1)/2/PI
      WRITE *,"[illegible]"
      GO TO 1[illegible]
1[illegible] CONTINUE
      A(L,D1)=A(L,D1)+(-ALD1*IL([illegible])-BLD1*IL([illegible])-ELD1*IL([illegible])+GTD1*IT(1)
     1+HTD1*IT(2)+2*ITD1*IT(3)-ATD1*IT([illegible])-BTD1*IT([illegible])+3*KTD1*IT([illegible])
     2-ETD1*IT([illegible])+3*JTD1*IT([illegible])/2)/2/PI
      A(L,D7)=A(L,D7)+(GLD7*IL(1)+HLD7*IL(2)+2*ILD7*IL(3)-ALD7*IL([illegible])
     1-BLD7*IL([illegible])+3*KLD7*IL([illegible])-ELD7*IL(7)+3*JLD7*IL([illegible])/2)/2/PI
      A(L,[illegible]7)=A(L,[illegible]7)+(-ALD7*IL([illegible])-BLD7*IL([illegible])-ELD7*IL(7))/2/PI
      WRITE *,"[illegible]"
1[illegible] CONTINUE
      A(L,G1)=A(L,G1)+(GLG1*IL(1)+HLG1*IL(2)+2*ILG1*IL(3)-ALG1*IL([illegible])-
     1BLG1*IL([illegible])+3*KLG1*IL([illegible])-ELG1*IL([illegible])+3*JLG1*IL(8)/2+GTG1*IT([illegible])
     2+HTG1*IT(2)+2*ITG1*IT([illegible])-ATG1*IT([illegible])-BTG1*IT([illegible])+3*KTG1*IT([illegible])
     3-ETG1*IT(7)+[illegible]*IT([illegible]))/2/PI
      A(L,G7)=A(L,G7)+(GLG7*IL(1)+HLG7*IL(2)+2*ILG7*IL(3)-ALG7*IL(4)
     1-BLG7*IL([illegible])+3*KLG7*IL([illegible])-ELG7*IL([illegible])+3*JLG7*IL([illegible])/2)/2/PI
      A(L,G7)=A(L,G[illegible])+(GTG[illegible]*IT(1)+[illegible]*IT(2)+2*ITG[illegible]*IT(3)-ATG[illegible]*IT([illegible])
     1-BTG[illegible]*IT([illegible])+3*KTG[illegible]*IT([illegible])-ETG[illegible]*IT(7)+LTG[illegible]*IT([illegible]))/2/PI
      A(L,D[illegible])=A(L,D[illegible])+(-ALD[illegible]*IT([illegible])-BTD[illegible]*IT([illegible])-ETD[illegible]*IT(7))/2/PI
      A(L,D6)=A(L,D6)+(GLD6*IL(1)+HLD6*IL(2)+2*ILD6*IL(3)-ALD6*IL([illegible])
     1-BLD6*IL([illegible])+3*KLD6*IL([illegible])-ELD6*IL(7)+3*JLD6*IL([illegible])/2)/2/PI
```

```
      A(L,[illegible])=A(L,[illegible])+(-A[illegible]*IT([illegible])-[illegible]*IT([illegible])-[illegible]IT(7))/4/PI
      A(L,[illegible])=A(L,[illegible])+([illegible]*IT(2)+[illegible]*IT(2)+2*[illegible]*IT(3)-[illegible]*IT([illegible])
     1-[illegible]*IT([illegible])+[illegible]*IT([illegible])-[illegible]*IT(7)+[illegible]IT([illegible])/2)/4/PI
      A(L,[illegible])=A(L,[illegible])+([illegible]*IT(1)+[illegible]*IT(2)+2*[illegible]*IT(3)-[illegible]*IT([illegible])
     1-[illegible]*IT([illegible])+3*[illegible]*IT([illegible])-[illegible]*IT(7)+3*[illegible]IT([illegible])/2)/4/PI
      WRITE *,"[illegible]"
      IF(N[illegible].EQ.[illegible]) GO TO [illegible]
      A(L,[illegible])=A(L,[illegible])+([illegible]*IT(1)+[illegible]*IT(2)+2*[illegible]*IT(3)-[illegible]*IT([illegible])
     1-[illegible]*IT([illegible])+3*[illegible]*IT([illegible])-[illegible]*IT(7)+[illegible]*IT([illegible]))/4/PI
      A(L,[illegible])=A(L,[illegible])+([illegible]*IT(1)+2*[illegible]*IT(2)-[illegible]*IT([illegible])-[illegible]*IT(7))
     1/4/PI
      WRITE *,"11"
1[illegible]  CONTINUE
1[illegible]  CONTINUE
1[illegible]  CONTINUE
1[illegible]  CONTINUE
1[illegible]  CONTINUE
[illegible]  CONTINUE
      I=2
      L=[illegible]
      X1=X(2,1)
      X2=X(2,2)
      X3=X(2,3)
      X4=X(2,4)
      Y1=Y(2,1)
      Y2=Y(2,2)
      CALL [illegible] (X,[illegible],I)
      CALL [illegible] (X1,X2,X3,X4,Y1,Y2)
      A(L,[illegible])=[illegible]
      A(L,[illegible])=[illegible]
      A(L,[illegible]1)=[illegible]-2*[illegible]*X2
      A(L,[illegible])=[illegible]-2*[illegible]*X2
[illegible]  RETURN
      END
```

```
      SUBROUTINE COMT (M,N,NP)
      COMMON/BLOCKC/A(12 ,12 ),SS(12 ),D3R(5 ),SUM(5 ),ALPH(1 ),[illegible]
      COMMON/BLOCKL/ALD1,ALD2,ALD3,ALD7,ALG1,ALG2,ALG4,ALG6,ALG7,
     1              CLD1,CLD2,CLD3,CLD7,CLG1,CLG3,CLG4,CLG6,CLG7,
     2              ELD1,ELD2,ELD3,ELD7,ELG1,ELG3,ELG4,ELG6,ELG7,
     3              ILD3,ILD5,ILG1,ILG3,ILG4,ILG5,ILG7,
     4              JLD3,JLD5,JLG1,JLG3,JLG4,JLG5,JLG7,
     5              KLD3,KLD5,KLG1,KLG3,KLG4,KLG5,KLG7,
     6              GLD3,GLD5,GLG1,GLG3,GLG4,GLG5,GLG7,
     7              HLD3,HLD5,HLG1,HLG3,HLG4,HLG5,HLG7,
     8              LLG3,LLG4,LLG5,
     9              ATD1,ATD2,ATD4,ATD5,ATD6,ATG1,ATG2,ATG5,ATG3,
     A              CTD1,CTD2,CTD4,CTD5,CTD6,CTG1,CTG2,CTG5,CTG3,
     B              ETD1,ETD2,ETD4,ETD5,ETD6,ETG1,ETG2,ETG5,ETG3,
     C              ITD1,ITD2,ITD5,ITG1,ITG2,ITG5,ITG3,
     D              JTD1,JTD2,JTD5,
     E              KTD1,KTD2,KTD5,KTG1,KTG2,KTG5,KTG3,
     F              GTD1,GTD2,GTD5,GTG1,GTG2,GTG5,GTG3,
     G              HTD1,HTD2,HTD5,HTG1,HTG2,HTG5,
     H              LTG1,LTG2,LTG3
      REAL  ILD3,ILD5,ILG1,ILG3,ILG4,ILG5,ILG7,
     1      JLD3,JLD5,JLG1,JLG3,JLG4,JLG5,JLG7,
     2      KLD3,KLD5,KLG1,KLG3,KLG4,KLG5,KLG7,
     3      LLG3,LLG4,LLG5,
     4      ITD1,ITD2,ITD5,ITG1,ITG2,ITG3,ITG5,
     5      JTD1,JTD2,JTD5,
     6      KTD1,KTD2,KTD5,KTG1,KTG2,KTG3,KTG5,
     7      LTG1,LTG2,LTG3
      COMMON/BLOCKK/G1,G2,G3,G4,G5,G6,G7,G8,D1,D2,D3,D4,D5,D6,D7,D8
      INTEGER G1,G2,G3,G4,G5,G6,G7,G8,D1,D2,D3,D4,D5,D6,D7,D8
      REAL MACH
      COMMON/BLOCKA/Y([illegible]),Y([illegible],3),YC([illegible]),YC([illegible]),Z([illegible],2),CY,CX,CYA,CYB,
     1CXA,CXB,XCA([illegible]),YCA([illegible]),XCB([illegible]),YCB([illegible])
      COMMON/BLOCKT/NS,SSP[illegible],S(15),[illegible],D(1 ),[illegible],[illegible](1 ),B(1 ,3),MACH
      MP=[illegible]
      AMCH=SQRT(1-MACH**2)
      DO 1 [illegible] I=1,NP
C
C        BASE EQUATION NUMBER
C
      L=2*NP+4*I-3
C
C        FOUR EQUATION NUMBERS.
C
      L1=L
      L2=L+1
      L3=L+2
      L4=L+3
C
C        [illegible]-TEVE COORDINATES.
C
      X1=X(I,1)/AMCH
      X4=X(I,4)/AMCH
      X3=X(I,3)/AMCH
      X2=X(I,2)/AMCH
      X5=X(I,5)/AMCH
      Y1=Y(I,1)
```

```
      Y2=Y(I,2)
      Y3=Y(I,3)
C
C              COMPUTE APPLICABLE NODAL VALUES.
C
      CALL NODNUM (M,N,I)
C              COMPUTE GEOMETRIC COEFFICIENTS.
C
      CALL COEFF (X1,X2,X3,X4,Y1,Y2)
C
C              COMPUTE LEADING AND TRAILING EDGE CHECK PARAMETERS.
C
      N3=I-(I/ )*N+1
      N9=I-(I/N)*N
C
C              IF TIP PANEL GO TO TIP SECTION
C
      IF(I.GT.(N*-*)) GO TO 2[illegible]
      IF(N[illegible].[illegible].N[illegible]) GO TO 11
      Y4=Y(I+N,2)
      X1 =Y(I+M,3)/AMOH
C
      A(L1,G1)=(-GTD1-HTD1*X9-ITD1*Y3-JTD1*X9*Y3-KTD1*Y3**2)*(X3-X1)
     1/(Y2-Y1)
      A(L1,G3)=( LD3+HLD3*X9+ILD3*Y3+JLD3*X9*Y3+KLD3*Y3**2)*(X1-X3)
     1/(Y4-Y2)
      A(L2,G1)=(ALD1+DLD1*Y3+ELD1*Y3**2-ATD1+ITD1*X9-DTD1*Y3+2*KTD1*Y3
     1*Y3-ETD1*Y3**2+JTD1 Y3**2/2)*(X3-X1)/(Y2-Y1)
      A(L2,G3)=(ALD3-ILD3*X9+DLD3*Y3-2*KLD3*X9*Y3+ELD3*Y3**2-JLD3*Y9 *2
     1/2)    *(X1 -X3)/(Y4-Y2)
      A(L2,G3)=(ALD3+DLD3 Y3+ELD3*Y3**2)*(X3-X1)/(Y2-Y1)
      A(L3,G1)=(ALD1+DLD1*Y2+ELD1*Y2**2)*(X3-X1)/(Y2-Y1)
      A(L3,G3)=(ALD3-ILD3*X4+DLD3*Y2-2*KLD3*X4*Y2+ELD3*Y2**2-JLD3*Y4
     1**2/2)*(X1 -X3)/(Y4-Y2)
      A(L3,G1)=(ALD1+DLD1 Y2+ELD1*Y2**2)*(X3-X1)/(Y2-Y1)
      A(L4,G1)=(GTD1+HTD1 X4+ITD1*Y2+JTD1*X4*Y2+KTD1*Y2**2)*(X3-X1)/
     1(Y2-Y1)
      IF(I.GT.1) GO TO 12
      A(L1,G5)=(-GTD5-HTD5*X9-ITD5*Y3-JTD5*X9*Y3-KTD5*Y3**2)*(X3-X1)
     1/(Y2-Y1)/2
      A(L2,G5)=(-ATD5+ITD5*X9-DTD5*Y3+2*KTD5*X9*Y3-ETD5*Y3**2+JTD5*X9
     1**2/2)*(X3-X1)/(Y2-Y1)/2
      A(L4,G5)=(GTD5+HTD5*X4+ITD5*Y2+JTD5*X4*Y2+KTD5*Y2**2)*(X3-X1)
     1/(Y2-Y1)/2
      GO TO 12
  1[illegible] CONTINUE
      A(L1,G3)=(GLD3+HLD3*X9+ILD3*Y3+JLD3*X9*Y3+KLD3*Y3**2)
      A(L2,G3)=(ALD3-ILD3*X9+DLD3*Y3-2*KLD3*X9*Y3+ELD3*Y3**2-JLD3*X9
     1**2/2)
      A(L3,G3)=(ALD3+DLD3*Y3+ELD3*Y3**2)
      A(L3,G3)=(ALD3-ILD3*X4+DLD3*Y2-2*KLD3*X4*Y2+ELD3*Y2**2-JLD3*X4
     1**2/2)
      A(L3,G3)=(ALD3+DLD3*Y2+ELD3*Y2**2)
      IF(I.LT.(N+1)) GO TO 12
      A(L1,G1)=(-GTD1-HTD1*X9-ITD1*Y3-JTD1*X9*Y3-KTD1*Y3**2)
      A(L2,G1)=(ALD1+DLD1*Y3+ELD1*Y3**2-ATD1+ITD1*X9-DTD1*Y3+2*KTD1*X9
     1*Y3-ETD1*Y3**2+JTD1*X9**2/2)
```

```
      A(L3,G1)=(ALG1+CLG1*Y2+ELG1*Y2**2)
      A(L4,G1)=(GTG1+HTG1*X4+ITG1*Y2+JTG1*X4*Y2+KTG1*Y2**2)
1[illegible] CONTINUE
      A(L1,D5)=(GLD5+HLD5*X3+ILD5*Y3+JLD5*X3*Y3+KLD5*Y3**2)
      A(L1,G3)=A(L1,G3)+(-GTG3-HTG3*X3-ITG3*Y3-KTG3*Y3**2-LTG3*X3**2)
      A(L1,G3)=(GLG3+HLG3*X3+ILG3*Y3+JLG3*X3*Y3+KLG3*Y3**2+LLG3*X3**2)
      A(L1,G1)=A(L1,G1)+(GLG1+HLG1*X3+ILG1*Y3+JLG1*X3*Y3+KLG1*Y3**2
     1-GTG1-HTG1*X3-ITG1*Y3-KTG1*Y3**2-LTG1*X3**2)
      A(L1,G7)=A(L1,G3)+(GLG3+HLG3*X3+ILG3*Y3+JLG3*X3*Y3+KLG3*Y3**2+
     1LLG3*X3**2)
      A(L1,G7)=(GLG7+HLG7*X3+ILG7*Y3+JLG7*X3*Y3+KLG7*Y3**2)
      A(L2,D4)=(-ATD4-CTD4*Y2-ETD4*Y2**2)
      A(L2,D5)=(ALD5-ILD5*X3+CLD5*Y3-2*KLD5*X3*Y3+ELD5*Y3**2-JLD5*X3
     1**2/2)
      A(L2,G3)=(-ATG3-CTG3*Y2-ETG3*Y2**2)
      A(L2,G3)=A(L2,G3)+(-ATG3+ITG3*X4-CTG3*Y2+2*KTG3*X4*Y2-ETG3*Y2**2)
      A(L2,G6)=(ALG6-ILG6*X3+CLG6*Y3-2*KLG6*X3*Y3+ELG6*Y3**2-JLG6*X3
     1**2/2)
      A(L2,G1)=A(L2,G1)+(ALG1-ILG1*X3+CLG1*Y3-2*KLG1*X3*Y3+ELG1*Y3**2
     1-JLG1*X3**2/2-ATG1+ITG1*X4-CTG1*Y2+2*KTG1*X4*Y2-ETG1*Y2**2)
      A(L2,G3)=A(L2,G3)+(ALG3-ILG3*X3+CLG3*Y3-2*KLG3*X3*Y3+ELG3*Y3**2
     1-JLG3*X3**2/2)
      A(L2,G7)=A(L2,G7)+(ALG7-ILG7*X3+CLG7*Y3-2*KLG7*X3*Y3+ELG7*Y3**2
     1-JLG7*X3**2/2)
      A(L3,D4)=-1.
      A(L3,D5)=(ALD5-ILD5*X4+CLD5*Y2-2*KLD5*X4*Y2+ELD5*Y2**2-JLD5*X4
     1**2/2)
      A(L3,G6)=(ALG6-ILG6*X4+CLG6*Y2-2*KLG6*X4*Y2+ELG6*Y2**2-JLG6*X4
     1**2/2)
      A(L3,G1)=A(L3,G1)+(ALG1-ILG1*X4+CLG1*Y2-2*KLG1*X4*Y2+ELG1*Y2**2
     1-JLG1*X4**2/2)
      A(L3,G3)=A(L3,G3)+(ALG3-ILG3*X4+CLG3*Y2-2*KLG3*X4*Y2+ELG3*Y2**2
     1-JLG3*X4**2/2)
      A(L3,G7)=A(L3,G7)+(ALG7-ILG7*X4+CLG7*Y2-2*KLG7*X4*Y2+ELG7*Y2**2
     1-JLG7*X4**2/2)
      A(L4,G3)=A(L4,G3)+(GTG3+HTG3*X4+ITG3*Y2+KTG3*Y2**2+LTG3*X4**2)
      A(L4,G1)=A(L4,G1)+(GTG1+HTG1*X4+ITG1*Y2+KTG1*Y2**2+LTG1*X4**2)
      IF(I.LT.([illegible])) GO TO [illegible]
      A(L1,D2)=(-GTD2-HTD2*X4-ITD2*Y2-JTD2*X4*Y2-KTD2*Y2**2)
      A(L1,D3)=(-GTD3-HTD3*X4-ITD3*Y2-JTD3*X4*Y2-KTD3*Y2**2)
      A(L2,D2)=(-ATD2+ITD2*X4-CTD2*Y2+2*KTD2*X4*Y2-ETD2*Y2**2+JTD2*X4
     1**2/2)
      A(L2,D3)=(-ATD3+ITD3*X4-CTD3*Y2+2*KTD3*X4*Y2-ETD3*Y2**2+JTD3*X4
     1**2/2)
      A(L4,D2)=(GTD2+HTD2*X4+ITD2*Y2+JTD2*X4*Y2+KTD2*Y2**2)
      A(L4,D3)=(GTD3+HTD3*X4+ITD3*Y2+JTD3*X4*Y2+KTD3*Y2**2)
1[illegible] CONTINUE
      IF([illegible].EQ.[illegible]) GO TO 1[illegible]
      A(L1,G4)=(GLG4+HLG4*X3+ILG4*Y3+JLG4*X3*Y3+KLG4*Y3**2+LLG4*X3**2)
      A(L1,G2)=(-GTG2-HTG2*X4-ITG2*Y2-KTG2*Y2**2-LTG2*X4**2)
      A(L1,G8)=(-GTG8-ITG8*Y2-KTG8*Y2**2)
      A(L2,G2)=(-ATG2+ITG2*X4-CTG2*Y2+2*KTG2*X4*Y2-ETG2*Y2**2)
      A(L2,G4)=(ALG4-ILG4*X3+CLG4*Y3-2*KLG4*X3*Y3+ELG4*Y3**2-JLG4*X3
     1**2/2)
      A(L2,G8)=(-ATG8+ITG8*X4-CTG8*Y2+2*KTG8*X4*Y2-ETG8*Y2**2)
      A(L3,G4)=(ALG4-ILG4*X4+CLG4*Y2-2*KLG4*X4*Y2+ELG4*Y2**2-JLG4*X4
     1**2/2)
```

```
      A(L4,G2)=(GTG2+HTG2*X4+ITG2*Y2+KTG2*Y2**2+LTG2*X4**2)
      A(L4,G.)=-1.
      A(L4,G3)=(GTG3+ITG3*Y2+KTG3*Y2**2)
      GO TO 15.
2 3 CONTINUE
      IF(NT.NE.MC) GO TO 21(
      A(L1,G1)=(-GTD1-HTD1*X3-ITD1*Y3-JTD1*X3*Y3-KTD1*Y3**2)*(X3-X1)
     1/(Y2-Y1)
      A(L2,G1)=(AL D1+CL D1*Y3+EL D1*Y3**2-ATD1+ITD1*X3-CTD1*Y3+2*KTD1*X3
     1*Y3-JTD1*Y3**2+JTD1*X3**2/2)*(X3-X1)/(Y2-Y1)
      A(L2,G7)=(AL D7+CL D7*Y3+EL D7*Y3**2)*(X3-X1)/(Y2-Y1)
      A(L3,G1)=(AL D1+CL D1*Y2+EL D1*Y2**2)*(X3-X1)/(Y2-Y1)
      A(L3,G7)=(AL D7+CL D7*Y2+EL D7*Y2**2)*(X3-X1)/(Y2-Y1)
      A(L4,G1)=(GTD1+HTD1*X4+ITD1*Y2+JTD1*X4*Y2+KTD1*Y2**2)*(X3-X1)/
     1(Y2-Y1)
      GO TO 22
21C CONTINUE
      A(L1,D1)=(-GTD1-HTD1*X3-ITD1*Y3-JTD1*X3*Y3-KTD1*Y3**2)
      A(L1,D7)=(GL D3+HL D3*X3+IL D3*Y3+JL D3*X3*Y3+KL D3*Y3**2)
      A(L2,D1)=(AL D1+CL D1*Y3+EL D1*Y3**2-ATD1+ITD1*X3-CTD1*Y3+2*KTD1*X3
     1*Y3-JTD1*Y3**2+JTD1*X3**2/2)
      A(L2,D7)=(AL D3-IL D3*X3+CL D3*Y3-2*KL D3*X3*Y3+EL D3*Y3**2-JL D3*Y3
     1**2/2)
      A(L2,D7)=(AL D7+CL D7*Y3+EL D7*Y3**2)
      A(L3,D1)=(AL D1+CL D1*Y2+EL D1*Y2**2)
      A(L3,D7)=(AL D7-IL D3*Y4+CL D7*Y2-2*KL D3*X4*Y2+EL D7*Y2**2-JL D3*Y4
     1**2/2)
      A(L3,D7)=(AL D7+CL D7*Y2+EL D7*Y2**2)
      A(L4,D1)=(GTD1+HTD1*X4+ITD1*Y2+JTD1*X4*Y2+KTD1*Y2**2)
22C CONTINUE
      A(L1,D2)=(-GTD2-HTD2*X3-ITD2*Y3-JTD2*X3*Y3-KTD2*Y3**2)
      A(L1,D5)=(GL D5+HL D5*X3+IL D5*Y3+JL D5*X3*Y3+KL D5*Y3**2)
      A(L1,G5)=(-GTG5-HTG5*X3-ITG5*Y3-KTG5*Y3**2-LTG5*X3**2)
      A(L1,G1)=A(L1,G1)+(GLG1+HLG1*X3+ILG1*Y3+JLG1*X3*Y3+KLG1*Y3**2
     1-GTG1-HTG1*X3-ITG1*Y3-KTG1*Y3**2-LTG1*X3**2)
      A(L1,G7)=(GLG7+HLG7*X3+ILG7*Y3+JLG7*X3*Y3+KLG7*Y3**2)
      A(L2,D2)=(-ATD2-CTD2*Y3-ETD2*Y3**2)
      A(L2,D5)=(AL D5-IL D5*X3+CL D5*Y3-2*KL D5*X3*Y3+EL D5*Y3**2-JL D5*X3
     1**2/2)
      A(L2,D5)=(-ATD5-CTD5*Y3-ETD5*Y3**2)
      A(L2,G1)=A(L2,G1)+(ALG1-ILG1*X3+CLG1*Y3-2*KLG1*X3*Y3+ELG1*Y3**2
     1-JLG1*X3**2/2-ATG1+ITG1*Y3-CTG1*Y3+2*KTG1*X3*Y3-ETG1*Y3**2)
      A(L2,G5)=         (-ATG5+ITG5*X3-CTG5*Y3+2*KTG5*X3*Y3-ETG5*Y3**2)
      A(L2,G7)=A(L2,G7)+(ALG7-ILG7*X3+CLG7*Y3-2*KLG7*X3*Y3+ELG7*Y3**2
     1-JLG7*X3**2/2)
      A(L3,G1)=A(L3,G1)+(ALG1-ILG1*X4+CLG1*Y2-2*KLG1*X4*Y2+ELG1*Y2**2
     1-JLG1*X4**2/2)
      A(L3,G7)=A(L3,G7)+(ALG7-ILG7*X4+CLG7*Y2-2*KLG7*X4*Y2+ELG7*Y2**2
     1-JLG7*X4**2/2)
      A(L3,D4)=-1.
      A(L3,D5)=(AL D5-IL D5*Y4+CL D5*Y2-2*KL D5*X4*Y2+EL D5*Y2**2-JL D5*X4
     1**2/2)
      A(L4,G5)=         (GTG5+HTG5*X4+ITG5*Y2+KTG5*Y2**2+LTG5*X4**2)
      A(L4,G1)=A(L4,G1)+(GTG1+HTG1*X4+ITG1*Y2+KTG1*Y2**2+LTG1*X4**2)
      A(L1,D3)=(-GTD3-HTD3*X3-ITD3*Y3-JTD3*X3*Y3-KTD3*Y3**2)
      A(L2,D2)=(-ATD2+ITD2*X3-CTD2*Y3+2*KTD2*X3*Y3-ETD2*Y3**2+JTD2*X3
     1**2/2)
```

```
      A(L2,G3)=(-ATG3+ITG3*X9-CTG3*Y3+2*KTG3*X9*Y3-ETG3*Y3**2+JTG3*Y9
     1**2/2)
      A(L4,G2)=(GTG2+HTG2*X4+ITG2*Y2+JTG2*X4*Y2+KTG2*Y2**2)
      A(L4,G3)=(GTG3+HTG3*X4+ITG3*Y2+JTG3*X4*Y2+KTG3*Y2**2)
      IF(N0.EQ.40) GO TO 23
      A(L1,G2)=(-GTG2-HTG2*X9-ITG2*Y3-KTG2*Y3**2-LTG2*X9**2)
      A(L1,G3)=(-GTG3-ITG3*Y3-KTG3*Y3**2)
      A(L2,G2)=(-ATG2+ITG2*X9-CTG2*Y3+2*KTG2*X9*Y3-ETG2*Y3**2)
      A(L2,G3)=(-ATG3+LTG3*X9-CTG3*Y3+2*KTG3*X9*Y3-ETG3*Y3**2)
      A(L4,G2)=(GTG2+HTG2*X4+ITG2*Y2+KTG2*Y2**2+LTG2*X4**2)
      A(L4,G3)=(GTG3+ITG3*Y2+KTG3*Y2**2)
 23   CONTINUE
 17   CONTINUE
 1    CONTINUE
      DO 579 I=1,54
      WRITE(6,599)(A(I,J),J=1,9)
      WRITE(6,599)(A(I,J),J=10,18)
      WRITE(6,599)(A(I,J),J=19,27)
      WRITE(6,599)(A(I,J),J=28,36)
      WRITE(6,599)(A(I,J),J=37,45)
      WRITE(6,599)(A(I,J),J=46,54)
 599  FORMAT (9(2X,E12.5))
      WRITE(6,597)
 597  FORMAT(///)
 579  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOADS2 (IF,M,N,NPD)
C                   SUBROUTINE TO CALCULATE PRESSURE DISTRI-
C                   BUTIONS AND AERODYNAMIC COEFFICIENTS.
      COMMON/BLOCKA/X( , ),Y( , ),XC( ),YC( ),S( ,2),CY,CX,CYT,CYB,
     1CYA,CYB,XCA( ),YCA( ),XCB( ),YCB( )
      COMMON/BLOCKB/ , ,F(10), ,D(1 ), , (1 ),B(1 ,2),MACH
      COMMON/BLOCKC/ (12 ,12 ),S (12 ),D R( ),SUM( ),ALPHA(1 ),N
      COMMON/BLOCKE/ ,DLTOT(1 ),NPLOT
      COMMON/BLOCKK/ , , , , , ,G7, , 1,D2,D3,D4,D5,D6,D7,D
      REAL LL,JT,KL,KT,LL,LT,LI
      COMMON/BLOCKL/ALD1,ALD2,ALD3,ALD4,ALG1,ALG2,ALG3,ALG5,ALG7,
     1              BLD1,BLD2,BLD3,BLD7,BLG1,BLG2,BLG ,BLG5,BLG7,
     2              ELD1,ELD3,ELD4,ELD7,ELG1,ELG3,ELG ,ELG5,ELG7,
     3              ILD7,ILD ,ILG1,ILG3,ILG ,ILG ,ILG7,
     4              JLD3,JLD ,JLG1,JLG3,JLG ,JLG ,JLG7,
     5              KLD3,KLD ,KLG1,KLG3,KLG ,KLG ,KLG7,
     6              GLD3,GLD ,GLG1,GLG3,GLG ,GLG ,GLG7,
     7              HLD3,HLD ,HLG1,HLG3,HLG ,HLG ,HLG7,
     8              LLG3,LLG ,LLG5,
     9              ATD1,ATD2,ATD ,ATD3,ATD ,ATG1,ATG2,ATG ,ATG3,
     A              CTD1,CTD2,CTD ,CTD3,CTD ,CTG1,CTG2,CTG ,CTG3,
     B              ETD1,ETD2,ETD ,ETD3,ETD ,ETG1,ETG2,ETG5,ETG3,
     C              ITD1,ITD2,ITD ,ITG1,ITG2,ITG ,ITG3,
     D              JTD1,JTD2,JTD5,
     E              KTD1,KTD2,KTD ,KTG1,KTG2,KTG ,KTG3,
     F              GTD1,GTD2,GTD3,GTG1,GTG2,GTG ,GTG3,
     G              HTD1,HTD2,HTD ,HTG1,HTG2,HTG ,
     H              LTG1,LTG2,LTG3
      REAL ILD3,ILD ,ILG1,ILG7,ILG ,ILG3,ILG ,
     1     JLD3,JLD ,JLG1,JLG3,JLG ,JLG3,JLG7,
     2     KLD3,KLD ,KLG1,KLG3,KLG ,KLG3,KLG7,
     3     LLG3,LLG ,LLG ,
     4     ITD1,ITD2,ITD ,ITG1,ITG2,ITG3,ITG5,
     5     JTD1,JTD2,JTD ,
     6     KTD1,KTD2,KTD ,KTG1,KTG2,KTG3,KTG5,
     7     LTG1,LTG2,LTG3
      INTEGER G1,G2,G3,G ,G ,G ,G7,G ,D1,D2,D3,D ,D ,D ,D7,D
      DIMENSION DEL( ),G ( ),DPU( ),DPL( ),DLL( ),D ( )
      DIMENSION CHORD(1 ),YS(1 ),DLD(1 ),LIFTD(1 ),DELY(1 )
      REAL MACH,LIFT ,MOMENT
      G=1.
      AMCH=SQRT(1-MACH**2)
      ID=
C
C                   BEGIN PLOTTING ROUTINES.  WHEN NPLOT= , NO
C                   PLOTS ARE DRAWN.
      IF (NPLOT.EQ. ) GO TO 3
C                   PLOTTER INITIALIZATION
      CALL PLOTS ( , , )
C                   POSITION PEN AND ESTABLISH ORIGIN
      CALL PLOT (2., ,-3)
C                   THIS SECTION PLOTS THE WING PLANFORM, NUMBERS
C                   EACH PANEL, AND MARKS THE CONTROL POINT OF EACH.
C                   WHEN A FACTOR OF 1. IS USED, THE ... IS
C                   NORMALIZED TO ... A TEN INCH SPAN.
      CALL FACTOR (. )
      CALL SYMBOL (-. ,3.7 ,.21,13HWING SEMISPAN, . ,13)
```

```
      CALL SYMBOL (-.+1C,.25,.21,1HPLANFORM,3C.,9)
      CALL SYMBOL (-.10,0.2 ,.14,11HSEMISPAN = ,C .,12)
      CALL NUMBER (9.9,  =., .3C,SSPN,3 .,1)
      CALL SYMBOL (.25,-.1C,. 9C,5HNINS (OOT,C.,9)
      BOKIF=1../SSPN
C                    THIS LOOP DRAWS AND LABELS EACH OF NP PANELS.
      DO 2 K=1,NP
         CALL PLOT (X(K,1)*BOKIF,Y(K,1)*BOKIF,3)
         CALL PLOT (X(K,2)*BOKIF,Y(K,1)*BOKIF,2)
         CALL PLOT (X(K,4)*BOKIF,Y(K,2)*BOKIF,2)
         CALL PLOT (X(K,3)*BOKIF,Y(K,2)*BOKIF,2)
         CALL PLOT (X(K,1)*BOKIF,Y(K,1)*BOKIF,2)
         CALL SYMBOL (XC(K)*BOKIF,YC(K)*BOKIF,.0 3,3,0.,-1)
         CALL NUMBER (X(K,3)*BOKIF+. 7,Y(K,2)*BOKIF-.1C,.C53,
     1   FLOAT (K),C .,-1)
    2 CONTINUE
      CONTINUE
C                    SET CAMBER SLOPES EQUAL TO ZERO.
      DO 1  I=1,NP
      CBR(I)= .
    1 CONTINUE
C                    READ NA SETS OF ANGLE OF ATTACK AND
C                    CAMBER SLOPE DISTRIBUTION DATA.
      DO 3C  L=1,NA
      READ (5,*) ALPHA(L),NCCG,NPF,CY,CX
C
C                    NCCG IS THE CAMBER CHANGE PARAMETER.
C                    ENTER  TO READ A NEW CAMBER SLOPE DISTRI-
C                    BUTION OR ENTER 1 TO RETAIN THE PREVIOUS
C                    DISTRIBUTION.
C
C                    NPF IS THE PRESSURE OPTION PARAMETER.
C                    ENTER  TO USE THE EXACT ISENTROPIC
C                    EXPRESSION OR ENTER 1 TO USE THE LINEAR-
C                    IZED FORM.
C
C                    COMPUTE PRESSURES AT PANEL LOCATION CX
C                    AND CY. NOTE - PRESSURES MAY BE COMPUTED
C                    AT POINTS OTHER THAN THE CONTROL POINTS.
C
C                    CALCULATE PRESSURE EVALUATION POINTS.
      DO 1C  I=1,NP
      YC(I)=(1-CY)*Y(I,1)+CY*Y(I,2)
   1C XC(I)=CY*X(I,3)+(1-CY)*X(I,1)+CX*(CY*(X(I,4)-X(I,3))+(1-CY)*
     1(X(I,2)-X(I,1)))
      IF(NCCG.EQ.1) GO TO 2
      READ(5,*) (CB(I),I=1,NP)
    2 CONTINUE
C                    FORMULATE LINEARIZED FORM OF THE FLOW
C                    TANGENCY BOUNDARY CONDITION. SUM(I)
C                    REPRESENTS THE NORMALIZED Z COMPONENT OF
C                    VELOCITY FOR PANEL I.
      DO 3  I=1,NP
         DEGRAD=57.29578
         SUM(I)=SIN(ATAN(CB(I)/DEGRAD)-ALPHA(L)/DEGRAD)
    3 CONTINUE
      DO 31 I=1,NP
```

```
      SUM(NP+I)=SIN(ATAN(DZ(I)/DEGRAD)-ALPHA(I)/DEGRAD)
   71 CONTINUE
C                  PREMULTIPLY SUM ARRAY BY A TO OBTAIN THE
C                  UNKNOWN SINGULARITY STRENGTHS.
      DO 5  J=1,NP2
      DUM=0.
      NP3=2*NP-1
      DO 40 I=1,NP3
      DUM=DUM+A(J,I)*SUM(I)
   40 CONTINUE
      SG(J)=DUM
    5 CONTINUE
      LTOT=0.
C
C                  START LOOP FOR COMPUTING THE SINGULARITY
C                  STRENGTH AT THE CONTROL POINT.
C
      DO 500 I=1,NP
C                  RETRIEVE CONTROL POINTS AND PANEL CORNER
C                  POINTS. APPLY PRANDTL-GLAUERT TRANSFORM.
      XI=XC(I)/AMCH
      YI=YC(I)
      X1=X(I,1)/AMCH
      X2=X(I,2)/AMCH
      X3=X(I,3)/AMCH
      X4=X(I,4)/AMCH
      Y1=Y(I,1)
      Y2=Y(I,2)
      N3=I-(I/M)*M-1
      N9=I-(I/M)*M
C
C             CALCULATE NODE NUMBERS
C
      CALL NODNUM (M,N,I)
C
C             SET NODAL VALUES TO ZERO.
C
      GAM1=0.
      GAM2=0.
      GAM3=0.
      GAM4=0.
      GAM5=0.
      GAM6=0.
      GAM7=0.
      GAM8=0.
      DEL1=0.
      DEL2=0.
      DEL3=0.
      DEL4=0.
      DEL5=0.
      DEL6=0.
      DEL7=0.
      DEL8=0.
C             IF A WING TIP PANEL GO TO TIP SECTION.
      IF(I.GT.(NP-M)) GO TO 30
C
C             NODAL VALUES DETERMINED IN THIS SECTION
```

```
C          ARE FOR ALL NON TIP PANELS.
C
      IF(NA.NE.MD) GO TO 110
      Y4=Y(I+1,2)
      X1=Y(I+M,3)/ROOTCH
      DEL1=(X3-X1)/(Y2-Y1)*SG(G1)
      WRITE*,"DEL1",DEL1
      DEL3=(X1-Y3)/(Y4-Y2)*SG(G3)
      WRITE*,"DEL3",DEL3
      DEL7=(X3-X1)/(Y2-Y1)*SG(G7)
      WRITE*,"DEL7",DEL7
      IF(I.GT.1) GO TO 12
      DEL5=(X3-X1)/(Y2-Y1)/2*SG(G5)
      GO TO 12
  110 CONTINUE
      DEL3=SG(D3)
      WRITE*,"DEL3",DEL3
      DEL7=SG(D7)
      WRITE*,"DEL7",DEL7
      IF(I.LT.(M+1)) GO TO 120
      DEL1=SG(D1)
      WRITE*,"DEL1",DEL1
  120 CONTINUE
      DEL4=SG(D4)
      WRITE*,"DEL4",DEL4
      DEL6=SG(D6)
      DEL8=SG(D8)
      WRITE*,"DEL8",DEL8
      GAM1=SG(G1)
      WRITE*,"GAM1",GAM1
      GAM3=SG(G3)
      WRITE*,"GAM3",GAM3
      GAM5=SG(G5)
      WRITE*,"GAM5",GAM5
      GAM6=SG(G6)
      WRITE*,"GAM6",GAM6
      GAM7=SG(G7)
      WRITE*,"GAM7",GAM7
      IF(I.LT.(M+1)) GO TO 121
      DEL2=SG(D2)
      WRITE*,"DEL2",DEL2
      DEL5=SG(D5)
  121 CONTINUE
      IF(NA.EQ.MD) GO TO 400
      GAM2=SG(G2)
      WRITE*,"GAM2",GAM2
      GAM4=SG(G4)
      WRITE*,"GAM4",GAM4
      GAM8=SG(G8)
      WRITE*,"GAM8",GAM8
      GO TO 400
  300 CONTINUE
C
C          NODAL VALUES DETERMINED IN THIS
C          ARE FOR WING TIP PANELS.
C
      IF(NA.NE.MD) GO TO 310
```

```
      DEL1=(X3-X1)/(Y2-Y1)*SG(G1)
      WRITE*,"DEL1",DEL1
      DEL7=(X3-X1)/(Y2-Y1)*SG(G7)
      WRITE*,"DEL7",DEL7
      GO TO 320
  310 CONTINUE
      DEL1=SG(D1)
      WRITE*,"DEL1",DEL1
      DEL3=SG(D3)
      WRITE*,"DEL3",DEL3
      DEL7=SG(D7)
      WRITE*,"DEL7",DEL7
  320 CONTINUE
      GAM1=SG(G1)
      WRITE*,"GAM1",GAM1
      GAM7=SG(G7)
      WRITE*,"GAM7",GAM7
      GAM5=SG(G5)
      WRITE*,"GAM5",GAM5
      DEL4=SG(D4)
      WRITE*,"DEL4",DEL4
      DEL5=SG(D5)
      WRITE*,"DEL5",DEL5
      DEL8=SG(D8)
      WRITE*,"DEL8",DEL8
      DEL2=SG(D2)
      WRITE*,"DEL2",DEL2
      DEL6=SG(D6)
      IF(NP.EQ.MP) GO TO 400
      GAM2=SG(G2)
      WRITE*,"GAM2",GAM2
      GAM4=SG(G4)
      WRITE*,"GAM4",GAM4
  400 CONTINUE
C
C             COMPUTE NODAL GEOMETRIC COEFFICIENTS.
C
      CALL COEFF (X1,X2,X3,X4,Y1,Y2)
C
C             COMPUTE COEFFICIENTS REQUIRED FOR GAMMA.
C
      GL=GLD3*DEL3+GLD6*DEL6+GLG1*GAM1+GLG3*GAM3+GLG4*GAM4+GLG5*GAM5
     1+GLG7*GAM7
C
      HL=HLD3*DEL3+HLD6*DEL6+HLG1*GAM1+HLG3*GAM3+HLG4*GAM4+HLG5*GAM5
     1+HLG7*GAM7
C
      LI=ILD3*DEL3+ILD6*DEL6+ILG1*GAM1+ILG3*GAM3+ILG4*GAM4+ILG5*GAM5
     1+ILG7*GAM7
C
      JL=JLD3*DEL3+JLD6*DEL6+JLG1*GAM1+JLG3*GAM3+JLG4*GAM4+JLG5*GAM5
     1+JLG7*GAM7
C
      KL=KLD3*DEL3+KLD6*DEL6+KLG1*GAM1+KLG3*GAM3+KLG4*GAM4+KLG5*GAM5
     1+KLG7*GAM7
C
      LL=LLG3*GAM3+LLG4*GAM4+LLG5*GAM5
```

```
C
      GT=GTD1*DEL1+GTD2*DEL2+GTD3*DEL3+GTG1*GAM1+GTG2*GAM2+GTG3*GAM3
     1+GTG5*GAM6
C
      HT=HTD1*DEL1+HTD2*DEL2+HTD3*DEL3+HTG1*GAM1+HTG2*GAM2+HTG3*GAM3
C
      TI=ITD1*DEL1+ITD2*DEL2+ITD3*DEL3+ITG1*GAM1+ITG2*GAM2+ITG3*GAM3
     1+ITG8*GAM8
C
      JT=JTD1*DEL1+JTD2*DEL2+JTD3*DEL3
      KT=KTD1*DEL1+KTD2*DEL2+KTD3*DEL3+KTG1*GAM1+KTG2*GAM2+KTG3*GAM3
     1+KTG9*GAM6
      LT=LTG1*GAM1+LTG2*GAM2+LTG3*GAM5
C                 CALCULATE LIFT AND MOMENT INCREMENTS
C
      WRITE*,"YI",YI
      WRITE*,"YT",YI
      X3=X1+(X3-X1)*(YI-Y1)/(Y2-Y1)
      X5=X1+(X -X1)*(YI-Y1)/(Y2-Y1)
      X7=X2+(X4-X2)*(YI-Y1)/(Y2-Y1)
      WRITE*,"X3,X5,X7",X3,X5,X7
C
      IF(NF.NE.NC) GO TO 410
      XLE=XC
  410 CONTINUE
C
      CLL(I)=2*(GL+LT*YI+KL*YI**2)*(X5-X3)+(HL+JL*YI)*(X5**2-X3**2)
     1+2*LL*(X5**3-X3**3)/3
      CLL(I)=CLL(I)+2*(GT+TI*YI+KT*YI**2)*(X7-X5)+(HT+JT*YI)*(X7**2
     1-X5**2)+2*LT*(X7**3-X5**3)/3
      WRITE*,"CLL",CLL(I)
      CM4(I)=(GL+LI*YI+KL*YI**2)*(X5**2-X3**2)+2*(HL+JL*YI)*(X5**3-X3*
     1*3)/3+LL*(X5**4-X3**4)/2
      CM4(I)=CM4(I)+(GT+TT*YI+KT*YI**2)*(X7**2-X5**2)+2*(HT+JT*YI)*
     1(X7**3-X5**3)/3+LT*(X7**4-X5**4)/2-XLE*CLL(I)
      WRITE*,"CM4",CM4(I)
C
C                 DETERMINE IF UPPER OR LOWER PANEL VOR-
C                 TICITY EQUATIONS ARE TO BE USED
C
C                 CALCULATE THE X CO-ORDINATE OF THE PANEL
C                 MAIN DIAGONAL AT YI.
      XD=X1+(X4-X1)/(Y2-Y1)*(YI-Y1)
C                 CHECK XI TO DETERMINE IF IT LIES ABOVE
C                 OR BELOW THE MAIN DIAGONAL. NOTE - IF
C                 XI.GT.XD, THEN IT LIES BELOW THE MAIN
C                 DIAGONAL.
      IF(XI.GT.XD) GO TO 
C
C                 UPPER PANEL EQUATIONS ARE IN THIS SECTION.
      AL=ALD1*DEL1+ALD3*DEL3+ALD5*DEL5+ALD7*DEL7+ALG1*GAM1+ALG3*GAM3
     1+ALG4*GAM4+ALG5*GAM5+ALG7*GAM7
C
      BL=BLD1*DEL1+BLD3*DEL3+BLD5*DEL5+BLD7*DEL7+BLG1*GAM1+BLG3*GAM3
     1+BLG4*GAM4+BLG5*GAM5+BLG7*GAM7
C
      CL=CLD1*DEL1+CLD3*DEL3+CLD5*DEL5+CLD7*DEL7+CLG1*GAM1+CLG3*GAM3
```

```
      GO TO 529
520   CONTINUE
      WRITE (6,528)
528   FORMAT(/33X,*LINEAR*)
529   CONTINUE
      IF (MCCG.EQ.1) GO TO 534
      WRITE (6,531)
531   FORMAT (/2 X,*NEW CAMBER DISTRIBUTION*)
      GO TO 535
534   CONTINUE
      WRITE (6,533)
533   FORMAT (/19X,*PREVIOUS CAMBER DISTRIBUTION USED*)
535   CONTINUE
      N1=1
      DO 760 J=1,N
      DUM=S(J)+DY*(S(J+1)-S(J))
      WRITE (6,570) DUM
570   FORMAT(///3 X,*Y/S =*,F6.3// 5X,*X/C*,4X,*DELTACP*,5X,*DEL*,5X,*GAM*,
     15X,*CPU*,5X,*CPL*,5X,*DELCP*)
      DO 750 I=1,M
      DUM1=C(I)+DX*(C(I+1)-C(I))
      DUM2=CPL(N1)-CPU(N1)
      WRITE (6,580) DUM1,CP (N1),DEL(N1),GAM(N1),CPU(N1),CPL(N1),DUM2
580   FORMAT(3X,F6.3,3Y,F6.2,2X,F8.5,3(1X,F9.5),2X,F9.5)
      N1=N1+1
750   CONTINUE
760   CONTINUE
C
C                   THIS SECTION OF THE OUTPUT TABULATES THE
C                   LOADS SUMMARY WHICH INCLUDES CL AND CM.
C
      WRITE(6,790)
790   FORMAT(///29X,*LOADS SUMMARY*//15X,*Y/S*,5X,*X-CP*,3X,*CHORD*,
     15X,*CL*,5X,*CM*)
      NB1=NB-1
      N4=0
      N5=0
C                   THE K LOOP MOVES ACROSS BREAK POINT
C                   SECTIONS
      DO 790 K=1,NB1
         N1 =NL(K)
         N2 =NL(K+1)-1
         N3 =NL(K+1)
C                   THE J LOOP MOVES ACROSS THE CHORDS
C                   IN A BREAK POINT SECTION, AND FINDS
C                   THE LENGTH OF EACH CHORD
         DO 780 J=N1,N2
            N5=N5+1
            YS(N5)=S(J)+DY*(S(J+1)-S(J))
            DUM1=(YS(N5)-S(N1))/(S(N3)-S(N1))
            CHORD(N5)=(B(K,2)-B(K,1))*(1-DUM1)+DUM1*(B(K+1,2)-B(K+1,1))
            CL1=0.
            CM1=0.
C                   THE I LOOP MOVES DOWN PANELS ALONG A CHORD
            DO 770 I=1,M
               N4 =N4+1
               CL1=CL1+CLL(N4)
```

```
     1+ELG3*GAM4+ELG5*GAM5+ELG7*GAM7
      DEL(I)=AL-LL*XI+CL*YI-2*KL*XI*YI+EL*YI**2-JL*YI**2/2
      WRITE*,"DEL",DEL(I)
      GAM(I)=GL+HL*XI+LI*YI+JL*XI*YI+KL*YI**2+LL*XI**2
      WRITE*,"GAM",GAM(I)
      GO TO 5
 4    CONTINUE
C
      AT=ATD1*DEL1+ATD2*DEL2+ATD3*DEL3+ATD5*DEL5+ATD *DEL7+ATG1*GAM1
     1+ATG2*GAM2+ATG3*GAM3+ATG5*GAM5
C
      CT=CTD1*DEL1+CTD2*DEL2+CTD3*DEL3+CTD5*DEL5+CTD *DEL7+CTG1*GAM1
     1+CTG2*GAM2+CTG3*GAM3+CTG5*GAM5
C
      ET=ETD1*DEL1+ETD2*DEL2+ETD3*DEL3+ETD5*DEL5+ETD *DEL7+ETG1*GAM1
     1+ETG2*GAM2+ETG3*GAM3+ETG5*GAM5
      GAM(I)=GT+HT*YI+TI*YI+JT*XI*YI+KT*YI**2+LT*XI**2
      WRITE*,"GAM",GAM(I)
      DEL(I)=AT-TI*XI+CT*YI-2*KT*YI*YI+ET*YI**2-JT*XI**2/2
      WRITE*,"DEL",DEL(I)
 5    CONTINUE
 5    CONTINUE
C
C                     CALCULATE UPPER AND LOWER SURFACE CPS.
C
      IF(NPF.EQ.1) GO TO 5
C                     EXACT ISENTROPIC PRESSURE EXPRESSION.
      DUM1=2/G/MACH**2
      DUM2=(G-1)*MACH**2/2
      DUM3=G/(G-1)
      DO 5 1 I=1,NP
      DUM4=(SUM(I))**2+(GAM(I)/2)**2+(DEL(I)/2)**2
      CPU(I)=DUM1*((1-DUM2*(GAM(I)+DUM4))**DUM3-1)
      CPL(I)=DUM1*((1-DUM2*(-GAM(I)+DUM4))**DUM3-1)
 5    CONTINUE
      GO TO 5
 5    CONTINUE
C                     LINEARIZED PRESSURE EXPRESSION.
      DUM1=(1-MACH**2)
      DO 5 5 I=1,NP
      DUM2=DUM1*(GAM(I)/2)**2+(SUM(I))**2+(DEL(I)/2)**2
      CPU(I)=-(GAM(I)+DUM2)
      CPL(I)=-(-GAM(I)+DUM2)
 5    CONTINUE
 5    CONTINUE
C
C                     START OUTPUT SEQUENCE
C
      WRITE (6,51 ) ALPHA(L)
 51   FORMAT (///  X,  (1H ),///,23X,*ALPHA =*,F5.2)
      WRITE(6,52 )
 52   FORMAT(//1 X,*PRESSURE AND SINGULARITY STRENGTH DISTRIBUTION*)
      WRITE (6,52 ) XY,CY
 52   FORMAT(/  X,*XP =*,F .3/3 X,*YP =*,F5.3)
      IF(NPF.EQ.1) GO TO 527
      WRITE (6,52 )
 52   FORMAT(/ 3X,*EXACT*)
```

```
              BB=LIFTC(J)
              CC=(YS(J+1)-YS(I))*SSPN
              DD=LIFTC(J+1)
              MOMENT=MOMENT+(CC**2*(DD-AA+2*BB)/3+(CC+AA)/6*(CC*BB-
     1        AA*DD)
              SHEAR=SHEAR+.5*DELY(J)*(BB+DD)
 32        CONTINUE
           WRITE (6,630) YS(I),SHEAR,MOMENT
 37     FORMAT (  X,F .2,-X,F7.1,3X,F .1)
 35     CONTINUE
C
C
C                    THIS SECTION PLOTS SECTIONAL LIFT COEFF VS WING
C                    SPANWISE STATION.  FOR THE FIRST LOADING CASE
C                    (WHEN L=1), THE AXES AND TITLE ARE DRAWN.  WHEN
C                    FACTOR =1., THE HORIZ AND VERT AXES ARE   IN AND
C                    1   IN, RESP.  WHEN NPLOT=0, NO PLOTS ARE DRAWN.
        IF (NPLOT.EQ.0.) GO TO
        IF (L.NE.1) GO TO
        CALL PLOT (2.,  .,-3)
        CALL FACTOR (.  )
C                    IMPOSE MIN AND SCALE FACTOR FOR YS
        YS(NS+1)=0.
        YS(NS+2)=1./ .
        CALL PLOT ( .,  .,3)
C                    THIS LOOP ACCOMPLISHES THE SAME AS SUBROUTINE
C                    AXIS, EXCEPT THAT IT ALLOWS AN   INCH AXIS TO
C                    BE DIVIDED INTO .1 INCREMENTS, AND HAVE A FULL
C                    SCALE VALUE OF 1. .
        DO 39  I1=1,11
           R=FLOAT(I1)-1.
           CALL PLOT (R*.3, .,3)
           CALL PLOT (R*.3,-. ,2)
           CALL NUMBER (R*.3-.125,-.22,.112,R*.1, .,2)
 39     CONTINUE
        CALL SYMBOL (2.125,-.35 ,.1  ,27HWING SPANWISE STATION (Y/B),
       1 .,27)
        CALL SCALE (CLC,1 .,NS,1)
        CALL AXIS ( ., .,19HSECTION LIFT COEFF.,19,1 .,9 .,CLC(NS+1),
       1CLC(NS+2))
        CALL SYMBOL (1.2 ,  .,.25,22HWING LOAD DISTRIBUTION, .,22)
 39     CONTINUE
C                    SUBROUTINE FLINE PLOTS A FAMILY OF CURVES, ONE
C                    CURVE EACH LOOP
        CALL FLINE (YS,CLC,-1*NS,1,1,1)
        CALL NUMBER (YS(1)/YS(NS+2)+.3,CLC(1)/CLC(NS+2)-.2 ,.125,
       1ALPHA(L), .,1)
        CALL SYMBOL (  .,  .,.12 ,11H CL,TOTAL= , .,11)
        CALL NUMBER (  .,  .,.12 ,CLTOT(L), .,4)
 3      CONTINUE
C                    CALCULATE CL,TOTAL VS ALPHA SLOPE, PROVIDING
C                    NA IS GREATER THAN 1
        IF (NA.LT.1) GO TO
        CLALPH=(CLTOT(NA)-CLTOT(1))/(ALPHA(NA)-ALPHA(1))*DEGRAD
        WRITE (6,9  ) CLALPH
 39     FORMAT (//1 X,  (1H ),//2 X,'D(CL)/D(ALPHA) = ',F .3,'   /RAD')
C
```

```
                 CM1=CM1+CNM(NS)
  77          CONTINUE
              XCP=CM1/(CL1*CHORD(NS))
              CLC(NS)=CL1/CHORD(NS)
              CM=CM1/(CHORD(NS)**2)
              WRITE(6,77) YS(NS),XCP,CHORD(NS),CLC(NS),CM,NS
  77          FORMAT(10X,F5.2,7X,F5.3,6X,F7.2,2X,F7.3,6X,F6.3,I4)
  78    CONTINUE
  79    CONTINUE
C                    WINGTIP CONDITIONS
C                    NOTE CHORD LENGTH AT WINGTIP IS AN ARBITRARY
C                    NO., SINCE MULTIPLIED BY ZERO CLC
      YS(NS)=1.
      CLC(NS)= .
      CHORD(NS)=1.
      WRITE(6,781) YS(NS), CLC(NS),NS
  781   FORMAT(10X,F5.2,14X,F7.3,9X,I4)
C
C                    THIS SECTION FINDS THE WING TOTAL LIFT
C                    COEFFICIENT BY NUMERICALLY INTEGRATING
C                    THE LOCAL LIFT COEFF. TIMES LOCAL CHORD
C                    (=LOCAL LIFT/DY AND PRESSURE) VS. SPANWISE
C                    DISTANCE CURVE.  THE INTEGRATION IS
C                    ACCOMPLISHED USING TRAPAZOIDAL AREAS.
C                    TOTAL CL IS THEN THIS VALUE DIVIDED BY WING
C                    AREA.
      AR1= .
      NS1=NS-1
      DO 795 I=1,NS
         LIFTO(I)=CLC(I)*CHORD(I)
         IF (I.EQ.NS) GO TO 795
         DELY(I)=(YS(I+1)-YS(I))*SSPN
  795   CONTINUE
      DO 797 I=1,NS1
         AR1=AR1+.5*(LIFTO(I)+LIFTO(I+1))*DELY(I)
  797   CONTINUE
C                    CALCULATE AREA UNDER CURVE FROM ROOT CHORD
C                    TO FIRST CONTROL POINT CHORD, ASSUMING A
C                    LINEAR CURVE FROM ROOT CL*CHORD TO CL(2)*CHORD(2)
      CL707=CLC(1)*CHORD(1)-((CLC(2)*CHORD(2)-CLC(1)*CHORD(1))/
     1(YS(2)-YS(1)))*YS(1)
      AR2=.5*YS(1)*SSPN*(CL707+CLC(1)*CHORD(1))
C                    TOTAL LIFT COEFF.
      CLTOT(L)=2.*(AR1+AR2)/AREA
      WRITE(6,800) CLTOT(L)
  800   FORMAT (//20X,*CL,TOTAL =*,F7.4)
C
C                    CALCULATE WING BENDING MOMENTS AS A FUNCTION
C                    OF SPANWISE STATION
      WRITE (6,831)
  831   FORMAT (///,30X,*WING BENDING*,//21X,*Y/S*,5X,*SHEAR/Q*,
     15X,*MOMENT/Q*)
      DO 840 I=1,NS1
         SHEARQ= .
         MOMENQ= .
         DO 820 J=I,NS1
            AA=(YS(J)-YS(I))*SSPN
```

```
C                 THIS SECTION PLOTS CL,TOTAL VS ALPHA, IF APPLICABLE
      IF (NPLOT.EQ.0) GO TO 91
      CALL PLOT (15.,[illegible].,-3)
      CALL FACTOR (.75)
      ALPHA(NA+1)=-1[illegible].
      ALPHA(NA+2)=[illegible]./3.
      CALL SCALE (CLTOT,[illegible].,NA,1)
      CALL AXIS ([illegible].,[illegible].,12HALPHA (DEG),-12,8.,[illegible].,ALPHA(NA+1),ALPHA(NA+2))
      CALL AXIS ([illegible].,[illegible].,[illegible]HCL,TOTAL,[illegible],[illegible].,[illegible].,CLTOT(NA+1),CLTOT(NA+2))
      CALL SYMBOL (.[illegible],[illegible].[illegible],.2[illegible],[illegible] LIFT COEFF.    ANGLE, [illegible].,2 )
      CALL SYMBOL (7.7[illegible],[illegible].25,.132,[illegible],[illegible].,3)
      CALL SYMBOL (2.1[illegible],[illegible],.2[illegible],[illegible]HOF  STACK,[illegible].,9)
      CALL SYMBOL (1.,[illegible].2[illegible],.12[illegible],1[illegible], [illegible].,-1)
      CALL SYMBOL ([illegible]3.,[illegible]35.,.12[illegible],[illegible]H(CL,TOTAL)/, [illegible].,11)
      CALL SYMBOL ([illegible]3.,[illegible]9.,.12[illegible],1[illegible], [illegible].,-1)
      CALL SYMBOL ([illegible]3.,[illegible],.12[illegible],[illegible]H(ALPHA) = , [illegible].,1 )
      CALL NUMBER ([illegible]3.,[illegible]5.,.12[illegible],CLALPH,[illegible].,3)
      CALL SYMBOL ([illegible]3.,[illegible]5.,.12[illegible],[illegible]/RAD.,[illegible].,4)
      CALL FLINE (ALPHA,CLTOT,-1*NA,1,1,1)
C
C                 PLOTTER TERMINATION ROUTINE.
      CALL PLOTE()
   91 CONTINUE
      RETURN
      END
```

Vita

Ronald E. Luther was born in Waukegan, Illinois on 11 June 1948. He was graduated from Chillicothe High School, Chillicothe, Ohio in 1966. He graduated from Purdue University with a Bachelor of Science degree (Aeronautical Engineering) in June 1970 and was commissioned into the Air Force at that same time. After completing Undergraduate Pilot Training he was assigned to the F-111 weapons system and was stationed at Nellis AFB, Nevada, Taklhi Royal Thai AFB, Thailand, and RAF Upper Heyford, England. In 1979 he entered the Air Force Institute of Technology.

Permanent Address: 4801 Sparrow Dr
Dayton, Oh 45424

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GAE/AA/80D-14	2. GOVT ACCESSION NO. AD-A094 775	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A HIGHER-ORDER TRAPEZOIDAL VECTOR VORTEX PANEL FOR SUBSONIC FLOW		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ronald E. Luther Captain, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 12 December 1980
		13. NUMBER OF PAGES 93
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

Fredric C. Lynch, Major, USAF
Director of Public Affairs 30 DEC 1980

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Airloads	Perturbation Method
Aerodynamic Loading	Paneling Method
Potential Flow	Vortex Panel
Subsonic Flow	Singularity Method

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A higher-order trapezoidal vector vortex panel method is developed for application to linearized subsonic potential flow. Each panel is subdivided into two triangular subregions on which a quadratic vorticity strength distribution is prescribed for both the spanwise and chordwise components of the vorticity vector. The vorticity strength distribution is

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20:

expressed as a function of the components of the vorticity vector at selected nodes on the boundary of each triangular subregion. Nodal values on the shared boundary of the subregions are made equal, assuring continuity of the vorticity distribution function throughout the trapezoidal panel. A lifting surface of no thickness is modeled with a network of the trapezoidal panels. Again, nodal values on the common panel boundaries are matched to achieve complete continuity of the vorticity distribution throughout the lifting surface. Aerodynamic data for several wing planforms is obtained with the flow model. Results from this method are compared to those from other computational and theoretical methods.